



Récepteur itératif pour système à multi-antennes basé sur l'algorithme de propagation de croyance

Ali Haroun

► To cite this version:

Ali Haroun. Récepteur itératif pour système à multi-antennes basé sur l'algorithme de propagation de croyance. Sciences de l'ingénieur [physics]. Télécom Bretagne; Université de Bretagne Occidentale, 2014. Français. NNT : . tel-01192808

HAL Id: tel-01192808

<https://hal.science/tel-01192808>

Submitted on 3 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE / Télécom Bretagne

sous le sceau de l'Université européenne de Bretagne

pour obtenir le grade de Docteur de Télécom Bretagne

En accréditation conjointe avec l'Ecole doctorale Sicma

Mention : Sciences et Technologies de l'Information et de la Communication

présentée par

Ali Haroun

préparée dans le département Electronique

Laboratoire Labsticc

Thèse soutenue le 21 novembre 2014

Devant le jury composé de :

Emmanuel Bouffillon

Professeur, Université de Bretagne-Sud / président

Charly Poulliat

Professeur, INP-ENSEEIH - Toulouse / rapporteur

Jean-Pierre Cances

Professeur, Ecole Nationale Supérieure d'Ingénieurs de Limoges / rapporteur

Matthieu Arzel

Maître de conférences, Télécom Bretagne / examinateur

Charbel Abdelnour

Maître de conférences, Télécom Bretagne / examinateur

Christophe Jegou

Professeur, IPB/ENSEIRB-MATMECA - Talence / directeur de thèse

Récepteur itératif pour système à multi-antennes basé sur l'algorithme de propagation de croyance

Sous le sceau de l'Université européenne de Bretagne

Télécom Bretagne

En accréditation conjointe avec l'Ecole Doctorale Sicma

Récepteur itératif pour système à multi-antennes basé sur l'algorithme de propagation de croyance

Thèse de Doctorat

Mention : *Sciences et technologies de l'information et de la communication*

Présentée par

Ali Haroun

Département : *Electronique*

Soutenue le : 21-11-2014

Jury :

Président du jury : **Emmanuel BOUTILLON**
Professeur des Universités, Université de Bretagne Sud

Rapporteurs : **Charly POULLIAT**
Professeur des Universités, INP - ENSEEIHT
Jean-Pierre CANCES
Professeur des Universités, ENSIL

Examineurs : **Matthieu ARZEL**
Maître de conférences, Télécom Bretagne

Charbel ABDELNOUR
Maître de conférences, Télécom Bretagne
Directeur de thèse : **Christophe JEGO**
Professeur des Universités, INP/ENSEIRB-MATMECA

Remerciements

Les travaux présentés dans ce manuscrit se sont déroulés au sein du département électronique à Télécom Bretagne et au sein du laboratoire CNRS-IMS.

Je souhaite remercier tout d'abord Christophe Jégo pour son rôle en tant que directeur de thèse ainsi que Matthieu Arzel et Charbel Abdelnour qui ont accepté d'encadrer cette thèse et qui m'ont accompagné et soutenu pendant toute la durée des travaux.

Je tiens à remercier vivement Emmanuel Boutillon professeur à l'UBS pour avoir présidé le jury de thèse ainsi que pour toutes les discussions intéressantes que nous avons eues. Je remercie également Charly Poulliat professeur à INP Toulouse et Jean-Pierre CANCES professeur à ENSIL Limoges pour avoir accepté de lire et d'évaluer mes travaux de thèse. La pertinence de leurs remarques atteste du soin avec lequel ils ont lu le manuscrit.

Je remercie les membres du département électronique à Télécom Bretagne ainsi que Michel Jézéquel, chef du département électronique, qui a permis que mes travaux se déroulent dans les meilleures conditions. Je remercie également tous les membres de l'équipe CSN au sein du laboratoire IMS à Bordeaux pour leur accueil chaleureux et leur bonne humeur.

Enfin, j'adresse toute mon affection à ma famille et particulièrement à mes parents, pour m'avoir soutenu tout au long de ma scolarité.

Enfin, je voudrais particulièrement remercier ma femme Ayate et mon fils pour leur amour, leur patience, leur dédicace et leur soutien toute la durée de cette thèse. Ayate a supporté mes doutes, mes moments d'absence et mes sauts d'humeur dans la forme la plus chaleureuse. Il n'existe personne comme elle sur cette planète. Merci, merci.

Résumé

Cette thèse présente une solution conjointe de détection et de décodage pour un système multi-antennes basé sur l'algorithme de propagation de croyance. Ce principe de propagation de croyance est appliqué aussi bien à la détection des symboles reçus qu'à leur décodage. Pour ce faire, un code correcteur d'erreurs de type LDPC (*Low-Density Parity-Check*) non-binaire est exploité. Nous proposons un nouveau schéma de réception itérative qui combine un détecteur MIMO (*Multiple-input Multiple-Output*) associé à un demapping avec une constellation d'ordre élevé suivi d'un décodeur LDPC non-binaire. Ce détecteur sous-optimal basé sur l'algorithme de propagation de croyance est étudié comme une alternative à un détecteur optimal basé sur le maximum de vraisemblance. L'information extrinsèque est échangée entre le détecteur et le décodeur au sein d'un processus itératif. La méthode des diagrammes EXIT (*EXtrinsic Information Transfer*) non-binaires a été appliquée afin d'analyser la convergence d'un récepteur itératif. La détection MIMO-BP et le décodage NB-LDPC sont combinés au sein d'un GFC (Graphe à Facteur Conjoint). Ce graphe peut être parcouru selon deux séquençements : le séquençement *flooding* et le séquençement *shuffle*. Dans le cadre de cette étude, ces deux types de séquençement sont comparés selon les critères de performance de correction, de latence et de complexité calculatoire.

Le principal objectif de ce travail de thèse est de réduire la complexité calculatoire du récepteur pour permettre son implémentation. Une première étape porte sur la mise à jour des distances euclidiennes du détecteur par les messages *a priori* provenant du décodeur. Ainsi, nous proposons deux techniques. La première technique porte sur la mise à jour des symboles les plus fiables par rapport au calcul initial des distances euclidiennes du détecteur. La seconde concerne la mise à jour des symboles par rapport aux informations les plus fiables provenant du décodeur. Ces deux propositions permettent une réduction importante de la complexité calculatoire tout en limitant les dégradations au niveau des performances. Dans un second temps, nous proposons une méthode de détection permettant de simplifier le calcul initial des distances euclidiennes. Cette dernière a des performances proches de celles du récepteur initial. Cette méthode consiste à déduire un certain nombre de points à partir du point le plus proche du symbole reçu. Une étude comparative en termes de performance et de complexité calculatoire des détecteurs simplifiés avec le détecteur original montre que les détecteurs proposés ont des caractéristiques favorables à une intégration matérielle au sein d'un système de réception MIMO itératif grâce une complexité divisée par 10 par rapport à la solution de référence.

Enfin, nous proposons une découpe architecturale pour les différents modules constituant le récepteur itératif. L'architecture proposée se compose d'un premier ensemble qui correspond à l'architecture du détecteur MIMO et d'un second pour le décodage du code LDPC-NB. Une représentation architecturale, une organisation temporelle et l'utilisation des ressources arithmétiques et de mémorisation sont détaillées pour les différents modules. Grâce à un niveau de parallélisme approprié, l'architecture associée au récepteur itératif permet un échange d'information efficace et par conséquent la montée en débit et la maîtrise de la latence du récepteur.

Mots-clés : Principe de la propagation de croyance, codes LDPC non-binaires, système à multi-antennes, graphe à facteur conjoint, information extrinsèque, récepteur itératif, diagramme EXIT.

Abstract

In this document, an efficient iterative receiver for digital communication systems is investigated. It combines a Non-Binary Low-Density Parity-Check (NB-LDPC) decoder with a high-order constellation demapper and a Multiple-Input Multiple-Output (MIMO) detector. In the first part of our work, we have proposed a suboptimal MIMO detector based on the Belief Propagation (BP) algorithm as an alternative to a Maximum Likelihood (ML) detector. Extrinsic information is exchanged between the detector and the decoder thanks to an iterative process. EXtrinsic Information Transfer (EXIT) charts have enabled to analyze the convergence behaviour of the proposed MIMO-BP detector. A suitable schedule for the different types of iterations is also proposed to reduce the receiver latency and improve its error correction performance. Indeed, a MIMO-BP detection and an NB-LDPC decoding can be combined together to form a larger Joint Factor Graph (JFG) which works on two different schedulings : a flooding scheduling and a Vertical Shuffle Scheduling (VSS). This approach enables to optimize the receiver schedule in terms of inter-iterations and intra-iterations to offer an efficient trade-off between performance and latency.

In a second part, we have focused on the main objective of this work : the reduction of the computational complexity of the iterative receiver to enable its implementation. Actions are undertaken at the detector, decoder and the iterative receiver levels in order to reduce overall computational complexity. The first step involved the updating of Euclidean distances within the detector by *a priori* messages from the decoder. Thus, we propose to act on two levels : the LDPC decoder level and the Euclidean distance update level. In the first level, a subset of Euclidean distance values is updated by decoder extrinsic information. This subset includes the symbol indexes with highest LLR reliability at the output of the detector during the previous inter-iteration. The second level is the update of the most reliable symbols relative to the initial calculation of the Euclidean distances of the detector. Indeed, applying the proposed low-complexity BP-based detection greatly reduces the number of operations per iteration, with a negligible performance penalty. In a second step, we have proposed a detection method to simplify the initial calculation of Euclidean distances. This method involves reducing a number of points around the closest points of the received symbol. We first calculate the first symbol candidate of the constellation, the remaining Euclidean distances are deduced recursively. This work enables to reduce the number of operations in the detector by a factor of ten when compared to full-complexity BP detection. The main interest of the proposed BP-based detector is to provide a feasible solution for the design of an iterative MIMO receiver.

In the last chapter, we propose a hardware architecture for the iterative receiver which consists in two parts. The first part is the architecture of the MIMO detector and the second part corresponds to the LDPC decoder. An architectural design, a time analysis and the complexity in terms of real operators are detailed for each block of the proposed architecture. Finally, the overall architecture proposed for the iterative receiver enables an efficient information exchange by an increase of the bandwidth and the reduction of the latency.

Index-Terms : Belief Propagation (BP), Non-Binary Low-Density Parity-Check (NB-LDPC), Multiple-Input Multiple-Output (MIMO), Joint Factor Graph (JFG), EXtrinsic Information Transfer (EXIT) chart, Iterative receiver.

Table des matières

1	Systèmes de Transmissions MIMO	1
1.1	Modèle d'un système mono-antenne	1
1.2	Notion de diversité	1
1.3	Transmission de l'information	2
1.3.1	Mapping/demapping	2
1.3.2	Canal de transmission	4
1.4	Codage et décodage canal	5
1.4.1	Les codes correcteurs d'erreurs	6
1.4.2	Les codes LDPC binaires	6
1.4.3	Les codes LDPC non-binaires	11
1.5	Les systèmes MIMO	17
1.5.1	Chaîne de transmission numérique de type MIMO	18
1.5.2	Capacité d'un système MIMO	19
1.6	Techniques de transmission des systèmes MIMO	21
1.6.1	Multiplexage temporel	21
1.6.2	Multiplexage spatial	21
1.6.3	Codage spatio-temporel par blocs	22
1.7	Techniques de détection à la réception de systèmes MIMO	23
1.7.1	Détection optimale MV (Maximum de Vraisemblance)	23
1.7.2	Détection MV bit par bit des systèmes MIMO	24
1.8	Performance du système MIMO 2x2 non itératif associé à un code LDPC non-binaire	25
1.8.1	Performances SISO vs MIMO	25
1.8.2	Performances système MIMO pour différents ordres de constellations	25
1.8.3	Performances système MIMO pour différents codes LDPC binaires et non-binaires	27
1.8.4	Performances système MIMO pour différentes techniques de transmissions MIMO	27
1.9	Conclusion	28
2	Récepteur itératif MIMO basé sur la propagation de croyance	29
2.1	Principe itératif à l'égalisation	29
2.2	Système de réception itératif	30
2.3	Représentation conjointe des graphes associés au détecteur et au décodeur	31
2.3.1	Détection bit par bit lors de la propagation de croyance	32
2.3.2	Détection non-binaire par propagation de croyance	35
2.3.3	Décodage non-binaire basé sur la propagation de croyance	38
2.4	Différents séquencements pour le système de réception	39
2.4.1	Séquencement <i>flooding</i>	39
2.4.2	Séquencement <i>shuffle</i>	40
2.5	Analyse du processus itératif du récepteur à l'aide de diagrammes EXIT	42
2.5.1	Les diagrammes EXIT au niveau binaire	42
2.5.2	Les diagrammes EXIT au niveau non-binaire	44
2.6	Etudes expérimentales	48
2.6.1	Performances du séquencement <i>flooding</i> : détection bit par bit versus détection symbole par symbole	48
2.6.2	Performances du séquencement <i>flooding</i> retenu à l'aide des diagrammes Exit au niveau non-binaire.	50

2.6.3	Performances du séquençement <i>shuffle</i> versus séquençement <i>flooding</i>	51
2.7	Conclusion	52
3	Algorithme de détection et de décodage conjoints à faible complexité calculatoire	53
3.1	Calcul récursif des distances euclidiennes par exploitation des points voisins de la constellation	53
3.2	Détecteur simplifié par Calcul Initial d'une Zone Réduite (CIZR) des distances euclidiennes	57
3.2.1	Détecteur simplifié par Région Statique (CIZR-S)	59
3.2.2	Détecteur simplifié par région dynamique (CIZR-D)	62
3.2.3	Etude comparative : détecteur de référence <i>versus</i> détecteur simplifié CI	69
3.3	Détecteur simplifié grâce à une Réduction du nombre d'Echanges d'Informations (REI) au cours de chaque inter-itération	71
3.3.1	Structure des messages tronqués	71
3.3.2	Compensation des vecteurs tronqués	75
3.3.3	Analyse de la complexité calculatoire et diagrammes EXIT pour l'algorithme de détection proposé	75
3.3.4	Etude comparative : détecteur de référence contre détecteur simplifié REI . . .	77
3.4	Détecteur simplifié basé sur les techniques CI et REI (CI-REI)	78
3.5	Conclusion	79
4	Architecture de réception MIMO itérative basée sur l'algorithme de propagation de croyance	81
4.1	Etat de l'art sur l'implémentation d'un détecteur MIMO	81
4.2	Architecture du récepteur MIMO itératif	82
4.3	Architecture du détecteur MIMO souple	86
4.3.1	Bloc CID (Calcul Initial des Distances euclidiennes)	87
4.3.2	Bloc Mémoire	98
4.3.3	Bloc AT (Additionneur-Tri)	99
4.3.4	Bloc AR (Additionneur-Registre)	100
4.4	Architecture du décodeur LDPC non-binaire	101
4.4.1	Architecture globale du décodeur et son séquençement	102
4.4.2	Système de mémorisation	103
4.4.3	Architecture du processeur associé aux nœuds de variable	104
4.4.4	Architecture du processeur associé aux nœuds de parité	107
4.5	Conclusion	112
	Bibliographie	119

Table des figures

1.1	Différents types de <i>mapping</i>	3
1.2	Modulation QAM à 64 états	3
1.3	Densité de probabilité de la loi de Rayleigh	5
1.4	Représentation graphique d'une équation de parité binaire pour trois bits.	7
1.5	Représentation de type graphe bipartite pour un code LDPC binaire.	8
1.6	Représentation d'un cycle de longueur 4.	9
1.7	Représentation de calcul d'un nœud de variable.	10
1.8	Représentation de calcul d'un nœud de parité.	10
1.9	Représentation matricielle d'un nœud de parité à contrainte non-binaire.	12
1.10	Représentation graphique associé au code LDPC non-binaire.	13
1.11	Représentation graphique associé au algorithme EMS.	15
1.12	Représentation graphique de la méthode récursive d'une équation de parité.	17
1.13	Représentation d'un système de transmission à un canal MIMO.	18
1.14	Représentation d'une chaîne de transmission de type MIMO	19
1.15	Capacités de différents systèmes MIMO et capacité d'un système SISO par rapport à la borne théorique de Shannon dans le cas d'un canal de Rayleigh	20
1.16	Multiplexage temporel de type H-BLAST	22
1.17	Multiplexage spatial de type V-BLAST	22
1.18	Performances du récepteur pour un canal SISO et un canal MIMO 2x2 de même efficacité spectrale.	26
1.19	Performances du récepteur pour un canal MIMO 2x2 avec les mappings QPSK, 16-QAM et 64-QAM	26
1.20	Performances système MIMO pour un code LDPC non-binaire avec différents longueurs de trame et pour un code LPDC binaire WIMAX.	27
1.21	Performances système MIMO pour différentes techniques de transmissions numériques.	28
2.1	Représentation du "principe turbo" ou "turbo-égalisation"	30
2.2	Chaîne de communications numériques étudiée	30
2.3	Représentation d'un graphe conjoint GFC	31
2.4	Etape 1 de la détection MIMO basé sur la propagation de croyance binaire	33
2.5	Etape 2 de la détection MIMO basé sur la propagation de croyance binaire	34
2.6	Récepteur itératif MIMO basé sur la propagation de croyance reposant sur un traitement symbole	36
2.7	séquencement <i>flooding</i> du GFC	39
2.8	Séquencement <i>shuffle</i> du GFC	40
2.9	Principe de mesure de l'information mutuelle entrante et sortante pour un détecteur MIMO.	43
2.10	Trajectoires des détecteurs MIMO pour un rapport RSB=11dB	44
2.11	Mesure de l'information mutuelle entrante et sortante pour un récepteur itératif.	45
2.12	Trajectoires des récepteurs MIMO pour un séquencement <i>flooding</i> pour un RSB=11 dB	46
2.13	Trajectoires des récepteurs MIMO pour différents séquencements avec un RSB=11 dB	47
2.14	Performances en termes TES d'une détection bit par bit	48
2.15	Performances en termes TES d'une détection symbole par symbole	49
2.16	Performances en termes TES de l'ordonnancement retenu à l'aide des diagrammes EXIT.	50
2.17	Performances du système de réception proposé selon différents séquencements	51

3.1	Illustration du calcul récursif des distances euclidiennes par exploitation des points voisins dans le cas d'une constellation 64-QAM.	56
3.2	Les quatre sous-régions statiques de la constellation 64-QAM	58
3.3	Les distances euclidiennes à deux dimensions pour les symboles S_1 et S_2	59
3.4	Le calcul par récursion des sous-régions statiques de la constellation 64-QAM	60
3.5	Performances de l'algorithme proposé CIZR-S par rapport à l'algorithme de détection originale	61
3.6	Traitement par sous-région dynamique versus statique	62
3.7	Système de communication MIMO 2x2	63
3.8	Détection du symbole S_1 le plus proche sachant que le symbole $S_2 = \alpha^k$	64
3.9	Calcul des distances euclidiennes du symbole S_1 sachant que $\{S_2 = \alpha^0, \alpha^1, \dots, \alpha^{63}\}$	67
3.10	Représentation des distances euclidiennes $D_{EUC}f_i$ en tenant compte des fiabilités	68
3.11	Performances de l'algorithme proposé CIZR-D par rapport à l'algorithme de détection CIZR-S	70
3.12	Tableau à deux dimensions des distances euclidiennes pour les combinaisons de S_1 et S_2 désignées par $\beta[y][x]$	72
3.13	Contribution portant sur le décodeur LDPC non-binaire	72
3.14	Contribution portant sur les calculs initiaux des distances Euclidiennes	73
3.15	Les sous-ensembles des distances euclidiennes à deux dimensions proposés pour la mise à jour.	74
3.16	Les diagrammes EXIT de l'algorithme BP à faible complexité pour différents profils.	76
3.17	Performances du système MIMO itératif pour différents jeux de paramètre pour l'algorithme BP.	78
3.18	Performances de l'algorithme proposé CI-REI par rapport à l'algorithme de détecteur BP original.	79
4.1	Architecture globale du récepteur MIMO	83
4.2	Découpe temporelle de la première inter-itération entre le détecteur et le décodeur	84
4.3	Découpe temporelle des autres inter-itérations, distinctes de la première, entre le détecteur et le décodeur	85
4.4	Architecture globale du récepteur MIMO itératif	87
4.5	Architecture du bloc Calcul Initial des Distances euclidiennes (CID)	88
4.6	Représentation architecturale du bloc 1 - $\{\widehat{y_1^{(I)}} \rightarrow P_1^I\}$	89
4.7	Architecture du bloc 1 - $\{\widehat{y_1^{(I)}} \rightarrow P_1^I\}$ pour le calcul du terme C	90
4.8	Architecture du bloc 1 - $\{\widehat{y_1^{(I)}} \rightarrow P_1^I\}$ pour calculer P_1^I	91
4.9	Organisation temporelle du bloc 1 - $\{\widehat{y_1^{(I)}} \rightarrow P_1^I\}$	91
4.10	Utilisation des ressources arithmétiques et de mémorisation du bloc 1 - $\{\widehat{y_1^{(I)}} \rightarrow P_1^I\}$	92
4.11	Découpe architecturale du bloc 2 - $\{D_{EUC}^I\}$	92
4.12	Organisation temporelle du bloc 2 - $\{D_{EUC}^I\}$	93
4.13	Utilisation des ressources arithmétiques et de mémorisation du bloc 2 - $\{D_{EUC}^I\}$	94
4.14	Découpe architecturale du bloc 3 - Tri	95
4.15	Organisation temporelle du bloc 3 - Tri	95
4.16	Découpe architecturale du bloc 4 - Calcul récursif	96
4.17	Organisation temporelle du bloc 4 - Calcul récursif	97
4.18	Assignment des ressources arithmétiques et de mémorisation pour le bloc 4 - Calcul récursif	97
4.19	Bloc Mémoire	98
4.20	Découpe architecturale du bloc AT	99

4.21	Organisation temporelle pour le bloc AT	100
4.22	Architecture du bloc AR	101
4.23	Illustration des étapes d'une intra-itération de décodage selon le séquençement <i>shuffle</i> à permutation verticale	101
4.24	Architecture globale du décodeur LDPC non binaire	102
4.25	Découpe architecturale pour un processeur associé aux nœuds de variable	104
4.26	Organisation temporelle pour la mise à jour du processeur associé aux nœuds de variable	106
4.27	Découpe architecturale du bloc décision	107
4.28	Matrice M_{at} associé à l'algorithme <i>L-Bubble check</i>	108
4.29	Architecture d'une étape élémentaire du processeur ECN associé aux nœuds de parité	109
4.30	Architecture d'un processeur associé aux nœuds de parité	110
4.31	Organisation temporelle de la mise à jour d'un processeur associé aux nœuds de parité	111
4.32	Organisation temporelle de l'ordonnancement itératif pour la première inter-itération du récepteur MIMO-BP	113
4.33	Organisation temporelle de l'ordonnancement itératif pour les autres inter-itérations du récepteur MIMO-BP	114

Liste des tableaux

2.1	Paramètres du système itératif MIMO basé sur la propagation de croyance	31
3.1	Coût calculatoire du terme $\min \mu $ pour un symbole reçu.	55
3.2	Comparaison au niveau complexité calculatoire entre la méthode classique et la méthode par récursion.	57
3.3	Le point initial et l'intervalle des points à calculer	60
3.4	Comparaison au niveau complexité calculatoire de la méthode CIZR-S par rapport aux autres méthodes proposées.	61
3.5	Comparaison au niveau complexité calculatoire de la méthode CIZR-D par rapport à la méthode CIZR-S.	69
3.6	Comparaison au niveau complexité calculatoire de la méthode CIZR-S par rapport aux autres méthodes proposées.	70
3.7	Comparaison au niveau des opérations (additions et comparaisons) entre différents détecteurs.	76
3.8	Comparaison au niveau des opérations (additions et comparaisons) entre différents paramètres pour le détecteur BP.	77
3.9	Comparaison au niveau des opérations de l'algorithme de détection de type CI-REI par rapport à l'algorithme de détection original.	79
4.1	Synthèse des caractéristiques des différents blocs pour le récepteur itératif	112
4.2	Comparaison des caractéristiques des différents détecteurs pour le récepteur itératif . .	114

Glossaire

AR	Addition Réelle
BP	Belief Propagation
Bloc AR	Additionneur-Registre
Bloc AT	Additionneur-Tri
BBAG	Bruit Blanc Additif Gaussien
CR	Registre de comparaison
CN	Check Node
CAM	Content Adressable Memory
CID	Calcul Initial des Distances euclidiennes
CIZR-S	Calcul Initial d'une Zone Réduite par région Statique
CIZR-D	Calcul Initial d'une Zone Réduite par région Dynamique
CI-REI	Calcul Initial basé sur la Réduction du nombre d'Echanges d'Informations
dB	déciBel
DAVINCI	Design And Versatile Implementation of Non-binary wireless Communications based on Innovative LDPC Codes
EMS	Extended Min-Sum
ECN	Elementary Check Node
EXIT	EXtrinsic Information Transfer
FPGA	Field Programmable Gate Array
FIFO	First Input First Output
GF	Galois Field
GFC	Graphe à Facteur Conjoint
IR	Inversion Réelle
Inter-itération	Itération globale entre le détecteur MIMO et le décodeur LDPC
Intra-itération	Itération interne au sein du détecteur MIMO ou au sein du décodeur LDPC-NB
LDPC	Low-Density Parity-Check
LDPC-NB	LDPC Non-Binaire
LRV	Logarithme du Rapport de Vraisemblance
MS	Min-Somme
MR	Multiplication Réelle
MV	Maximum de Vraisemblance
MIMO	Multiple-Input Multiple-Output
MIMO-BP	MIMO basé sur l'algorithme BP
MIMO-MV	MIMO basé sur l'algorithme MV
MMSE	Minimum Mean Square Error

QPSK	Quaternary Phase Shift keying
QAM	Quadrature Amplitude Modulation
RSB	Rapport Signal à Bruit
RAM	Random Access Memory
SISO	Single-Input Single-Output
SR	Soustraction Réelle
TES	Taux Erreur Symbole
VN	Variable Node
WIMAX	Worldwide Interoperability for Microwave Access

Notations

b_i	Vecteur des bits candidats de taille $N_t \times q$
C_r	$r^{\text{ème}}$ nœuds de parité
$CtoV$	Messages LRV envoyés du processeur CN au processeur VN
d_v	Degré de connectivité par nœud de parité
d_c	Degré de connectivité par nœud de variable
det	Nombre d'intra-itérations au sein du détecteur
dec	Nombre d'intra-itérations au sein du décodeur
det_{nc}	Contribution de n_c messages provenant du détecteur MIMO
dec_{nv}	Contribution de n_v messages provenant du décodeur LDPC non-binaire
D_{EUC}	Distance Euclidienne
$f(\lambda/x)$	La densité de probabilité conditionnelle
H	Matrice de parité du code LDPC
H_c	Matrice des coefficients du canal MIMO
$h_{j,i}^I$	Coefficient du canal de la $i^{\text{ème}}$ antenne de transmission à la $j^{\text{ème}}$ antenne de réception sur l'axe I de la constellation.
$h_{j,i}^Q$	Coefficient du canal de la $i^{\text{ème}}$ antenne de transmission à la $j^{\text{ème}}$ antenne de réception sur l'axe Q de la constellation.
I_A	Information mutuelle à l'entrée du détecteur et à la sortie du décodeur
I_E	Information mutuelle à l'entrée du décodeur et à la sortie du détecteur
I_{pversv}	Information mutuelle passant du $r^{\text{ème}}$ nœud de parité au $p^{\text{ème}}$ nœud de variable
I_{vversp}	Information mutuelle passant du $p^{\text{ème}}$ nœud de variable au $r^{\text{ème}}$ nœud de parité
$I(\lambda/x)$	Information mutuelle moyennée
K	Nombre de symboles d'information
L_{b_i}	LRV du bit candidat b_i
L_{c_i}	LRV du symbole candidat c_i
L_{V_p}	LRV du nœuds de variable V_p
$L_{p \rightarrow b_m}$	Information extrinsèque provenant du décodeur LDPC-NB envoyée au bit candidat b_m
$L_{pr \rightarrow S_i}$	Information extrinsèque provenant du décodeur LDPC-NB envoyée au symbole candidat S_i
$L_{y_j \rightarrow S_i}$	Messages LRV envoyés du symbole reçu y_j au symbole candidat S_i
$L_{S_i \rightarrow y_j}$	Messages LRV envoyés du symbole candidat S_i au symbole reçu y_j
$L_{[S_i; \alpha^k]}$	La fiabilité pour le symbole candidat S_i d'être égal à α^k

$L_{l(p,r) \rightarrow V_p}$	Informations souples envoyées du $r^{\text{ème}}$ nœuds de parité au $p^{\text{ème}}$ nœuds de variable
$L_{pr \rightarrow V_p}$	Informations souples associées aux symboles envoyées au décodeur LDPC-NB
M	Nombre de symboles de redondance
M_{CtoV}	Mémoire pour stocker les messages $CtoV$
M_{VtoC}	Mémoire pour stocker les messages $VtoC$
M_{det}	Mémoire pour stocker les messages provenant du détecteur
M_{at}	Matrice contient les nm_{dec}^2 candidats pour la mise à jour du vecteur provenant du VN
N	Longueur du mot de code
N_t	Nombre d'antennes à l'émission
N_r	Nombre d'antennes à la réception
NDD	Nombre d'inter-itérations entre le détecteur et le décodeur
nm_{dec}	Nombre de couples échangés entre le processeur VN et le processeur CN
n_{GF}	Nombre de bits codés pour un symbole défini sur un corps de galois GF
n_{LRV}	Nombre de bits codés pour le rapport de vraisemblance d'un symbole non-binaire
n_s	Nombre de couples sauvegardés dans la CAM
n_{oper}	Nombre d'opérations nécessaires à la construction de la liste des nm_{dec} symboles GF
q	Ordre de modulation
R	Rendement du code LDPC
S_i	Vecteur des symboles candidats de taille N_t
S_{oper}	Nombre de cycles d'exécution nécessaires pour générer les nm_{dec} symboles
V_p	$p^{\text{ème}}$ nœuds de variable
$VtoC$	Messages LRV envoyés du processeur VN au processeur CN
y_j	Vecteur des symboles reçus de taille N_r
α^k	Elément appartenant au corps du Galois non binaire $GF(2^6)$
λ	Information extrinsèque pondérée de l'élément binaire x
σ_a^2	La variance du canal virtuel de l'information mutuelle
σ_w^2	La variance du bruit BABG
σ_s^2	Facteur de normalisation d'une constellation M QAM
$\beta[i][ii]$	Distance Euclidienne tel que S_1 est égal au point de la constellation α_i et S_2 au point de la constellation α_{ii}

Introduction

Afin de répondre aux contraintes concernant le débit et la puissance d'émission, les réseaux de radio-communication se tournent vers des systèmes à plusieurs antennes à l'émission et à la réception. De tels systèmes MIMO (*Multiple-Input Multiple-Output*) permettent d'accroître la capacité des liens de communications par rapport aux systèmes SISO (*Single-Input Single-Output*) qui ne contiennent qu'une antenne à l'émission et une à la réception [1]. La recherche portant sur des systèmes de communications numériques MIMO s'est focalisée dans un premier temps sur l'établissement de leurs performances théoriques et sur la proposition d'ensembles émetteur/récepteur permettant de tirer bénéfice de l'utilisation de plusieurs antennes [2]. En effet, les systèmes MIMO s'affranchissent des évanouissements des canaux de transmission par une meilleure exploitation de la diversité.

En outre, l'association des techniques MIMO avec des codes correcteurs d'erreurs permet d'aboutir à des transmissions de grande fiabilité. Depuis l'introduction des turbocodes en 1993 [3], le traitement itératif appelé turbo-égalisation ou turbo-détection permet d'obtenir des gains importants en termes de Taux d'Erreur Symbole (TES), par rapport à un traitement non-itératif [4]. Malgré la multitude des travaux de recherche dans ce domaine et l'existence de plusieurs détecteurs MIMO [5] [6], la complexité du détecteur et la latence associée demeurent parmi les principaux axes d'amélioration possibles. C'est pourquoi les techniques de détection MIMO basées sur la propagation de croyance BP (*Belief Propagation*) apparaissent comme une solution offrant de forts potentiels pour une implémentation matérielle [7].

Les codes LDPC (*Low-Density Parity-Check*) binaires ont été définis pour la première fois par Gallager en 1962 [8]. Cependant, ces travaux n'ont pas fait l'objet d'études complémentaires pendant 30 ans. Suite à l'apparition des turbocodes et du traitement itératif, les codes LDPC furent redécouverts au milieu des années 1990. En 1998, une première version non-binaire a été proposée [9]. Elle était capable de surpasser les codes LDPC binaires en terme de performance, surtout pour des trames courtes [10]. Grâce à sa structure non-binaire, cette famille fut naturellement associée à des constellations d'ordre élevées [11]. Pour les mêmes raisons, elle fût également utilisée pour des systèmes multi-antennes MIMO à haut débit. C'est pourquoi la proposition d'un traitement itératif en réception basé sur un graphe de propagation conjoint entre le détecteur MIMO et le décodeur de code LDPC doit permettre la conception de récepteurs flexibles caractérisés par de faibles latences.

Problématique et Contributions

Des récepteurs MIMO itératifs sont difficiles à concevoir sur une cible matérielle. De plus, les architectures résultantes sont peu efficaces en termes de débit en raison des traitements itératifs et de l'exploitation de la dimension spatiale. Cependant, leur bon résultat au niveau des performances, par rapport aux récepteurs non itératifs, nous invite à penser que la conception de ces récepteurs itératifs est un défi scientifique intéressant à relever. Les systèmes MIMO itératifs permettent

d'accroître la capacité des liens de communication sans fil, grâce à l'apport de la diversité spatiale. Les techniques de détection MIMO basées sur la propagation de croyance offrent un avantage indéniable au niveau de la complexité calculatoire par rapport aux techniques de détection présentes dans la littérature pour une implémentation matérielle.

Ainsi, l'objectif de cette thèse est de modéliser une chaîne de transmission MIMO intégrant une détection basée sur la propagation de croyance suivie d'un décodage de codes LDPC non-binaires également basé sur la propagation de croyance. L'étude consiste à proposer des solutions algorithmiques et architecturales afin de maîtriser les problèmes de complexité calculatoire et de l'échange d'information au sein du récepteur itératif. Au niveau de la détection MIMO, l'exploitation de la dimension spatiale a un impact sur la complexité des détecteurs MIMO. Ainsi, la simplification de l'algorithme de détection et la proposition des solutions architecturales constituent les principaux défis à relever. Pour ce faire, une reformulation de l'algorithme de détection et une étude comparative au niveau performance/complexité calculatoire sont présentées. Dans un second temps, des solutions architecturales pour les différents modules constituant le récepteur itératif sont détaillées.

Plan du mémoire

Ce mémoire est organisé en quatre chapitres.

Le premier chapitre présente les concepts associés aux communications numériques et les notions essentielles pour aborder les autres chapitres du mémoire. Nous décrivons une chaîne de transmission numérique composée d'un codage/décodage de canal, d'une modulation/démodulation numérique et d'un canal de transmission. Puis, nous présentons les caractéristiques et les notions associées aux différents canaux de transmission. Les techniques de multi-antennes à l'émission et à la réception sont abordées par la suite. A la réception, la détection des informations reçues ainsi que le décodage des codes LDPC employés par la suite sont explicités. Notons que nous considérons dans cette étude des techniques qui supposent un canal inconnu à l'émission et estimé à la réception. En bref, le premier chapitre pose le cadre de cette étude et présente les différentes fonctionnalités utilisées dans le récepteur itératif considéré dans les chapitres suivants.

Le deuxième chapitre décrit un récepteur de communications numériques réalisant conjointement la détection d'un système MIMO et le décodage de codes LDPC au sein d'un processus itératif global. Dans un premier temps, nous présentons un schéma de réception MIMO admettant un détecteur sous-optimal basé sur l'algorithme de propagation de croyance. Dans un second temps, nous étudions des réceptions itératives reposant sur deux séquençements différents pour l'échange des messages entre le détecteur et le décodeur : "Séquencement *flooding*" et "Séquencement *shuffle*". De plus, nous exploitons des diagrammes EXIT (*EXtrinsic Information Transfer*) afin d'analyser le comportement de convergence du détecteur MIMO basé sur la propagation de croyance. Enfin, une analyse des performances et de complexité calculatoire du récepteur est faite.

Le troisième chapitre concerne la recherche d'algorithmes de détection et de décodage conjoints ayant une complexité calculatoire réduite. Nous diminuons la complexité calculatoire de notre récepteur itératif à deux niveaux. Dans un premier temps, nous réduisons le coût du calcul initial des distances euclidiennes pour le détecteur MIMO. Puis, nous traitons la réduction du nombre

d'échanges d'informations au cours du processus itératif. En outre, à l'aide de diagrammes EXIT, nous déterminons le jeu de paramètres le plus pertinent au niveau de l'échange d'informations au sein du récepteur itératif.

Sur la base des résultats obtenus précédemment, le quatrième chapitre porte sur la proposition d'une découpe architecturale pour le récepteur itératif. Nous proposons des solutions architecturales pour les différents modules constituant le récepteur itératif. L'architecture proposée se compose d'une première partie qui correspond à l'architecture du détecteur MIMO et d'une seconde pour l'architecture de décodage d'un code LDPC-NB. Tout d'abord, nous décrivons les différents blocs de l'architecture du détecteur MIMO. Puis, nous détaillons pour chaque bloc constituant le détecteur MIMO, sa représentation architecturale, son organisation temporelle et l'utilisation des ressources arithmétiques et de mémorisation. Dans un second temps, nous détaillons les différents blocs de l'architecture du décodeur LDPC non-binaire en considérant un seul processeur pour les nœuds de variable et un seul processeur pour les nœuds de parité. Nous aboutissons donc à une architecture globale pour le récepteur itératif permettant un échange efficace d'information par le biais d'un parallélisme favorisant la montée en débit et la maîtrise de la latence du récepteur.

1 Systèmes de Transmissions MIMO

Ce chapitre présente les bases des communications numériques utilisées dans cette thèse. Nous décrivons tout d'abord la chaîne de transmission numérique composée d'un codage de canal, d'une modulation et d'un canal de transmission. Différents canaux de transmission sont présentés : le canal à bruit additif blanc Gaussien, le canal de Rayleigh et de systèmes multi-antennes. A la réception, la détection des informations reçus ainsi que le décodage des codes LDPC employés par la suite sont détaillés. En bref, dans ce chapitre nous posons le cadre de notre travail et nous présentons les différentes notions utilisées dans les chapitres à venir.

1.1 Modèle d'un système mono-antenne

Une chaîne de transmission numérique permet de transmettre une quantité d'information numérique, avec le moins d'erreurs possibles, entre un émetteur et un récepteur à travers un canal de transmission. Ce canal possède un certain nombre de caractéristiques et de contraintes dont il faut tenir compte. Le cas classique et le plus simple pour des canaux de transmission est le système mono-antenne. Un canal mono-antenne ou SISO (*Single-Input Single-Output*) [12] se caractérise par une antenne d'émission et une antenne de réception.

Dans le cadre de notre étude, nous nous intéressons aux canaux radio-mobiles. Un canal radio mobile [13] mono-antenne peut être modélisé par un bruit additif blanc Gaussien et un bruit multiplicatif. Le bruit additif s'ajoute au signal transmis au récepteur. Le bruit multiplicatif [14] représente les évanouissements que subit le signal transmis au cours de différents chemins qu'il peut prendre pour parvenir au récepteur. Un canal radio-mobile est ainsi caractérisé par ses évanouissements. Dans cette étude, nous considérons que les évanouissements suivent une loi de Rayleigh [15] où chaque coefficient du canal est un nombre complexe Gaussien de moyenne nulle et de variance 1, indépendant et identiquement distribué (i.i.d).

1.2 Notion de diversité

En présence d'un évanouissement profond, transmettre une seule réplique du signal peut s'avérer insuffisant pour la récupération de l'information dans le cas d'une distribution de Rayleigh. Une solution est de posséder à la réception de différentes occurrences du signal affectées d'évanouissements indépendants pour s'assurer de récupérer le signal transmis lors de l'émission. Plus le nombre de canaux indépendants augmente, plus la probabilité que tous les canaux indépendants subissent des évanouissements profonds diminue. Cette propriété correspond à la notion de diversité.

La diversité est alors utilisée pour combattre les évanouissements. Cette diversité permet de sélectionner le meilleur canal, de combiner les signaux reçus afin de bénéficier de l'apport de chacun

et de mettre en forme les signaux à l'émission pour maximiser ce bénéfice. En pratique plusieurs types de diversité sont distinguées :

Diversité fréquentielle : Ce type de diversité consiste à envoyer le même signal sur des fréquences différentes séparées d'au moins la bande de cohérence du canal tout en respectant la bande de fréquence disponible. Cette technique de diversité peut être obtenue lorsque le canal est sélectif en fréquence et généralement utilisée dans les systèmes OFDM (*Orthogonal Frequency Division Multiplex*) [16], [17].

Diversité temporelle : Elle consiste à envoyer un même signal à des instants différents séparés d'au moins le temps de cohérence du canal afin d'assurer une bonne décorrélation des signaux. Cette technique de diversité est intéressante dans le cas des canaux à évanouissement rapide [18].

Diversité spatiale : Elle est aussi connue sous le nom de diversité d'antennes. Elle est apportée par la présence de plusieurs antennes à l'émission et/ou à la réception. Cette diversité consiste à envoyer le même signal sur des antennes différentes. Les signaux reçus sur chacune des antennes de réception sont ensuite remis en phase et sommés de façon cohérente [19].

Dans les systèmes de transmission actuels, afin d'obtenir des ordres de diversité ainsi que des efficacités spectrales élevées, plusieurs types de diversité sont combinées entre eux [20]. En effet, l'ordre total de diversité sera égal au produit des différents ordres de diversité.

1.3 Transmission de l'information

Généralement, le signal transportant une information doit être échangé via un milieu de transmission entre un émetteur et un récepteur. Le signal est rarement adapté à une transmission directe à travers un canal de communication. La modulation peut être définie comme le processus par lequel le signal est transformé de sa forme originale en une forme adaptée au canal de transmission choisi. La transmission de l'information consiste donc à acheminer de l'information provenant d'une source vers un destinataire, à travers un canal de transmission.

1.3.1 Mapping/demapping

Les modulations utilisées dans le contexte de cette étude font varier les paramètres de phase et d'amplitude d'une onde porteuse à partir d'une fréquence constante. Le signal émis est caractérisé par ces deux paramètres et représenté par des symboles complexes discrets. La constellation $\chi = (x_0, x_1, \dots, x_{M-1})$ contient un nombre de points M dont la valeur est une puissance de 2. Chaque groupe de $q = \log_2(M)$ données binaires est représenté par un symbole dans la constellation. La fonction de *mapping* associe à chaque groupe de bits $[b_0, b_1, \dots, b_{q-1}]$ un symbole complexe. On distingue quelques types de modulations comme :

- Déplacement de phase à 2 états BPSK (*Binary Phase-Shift Keying*)
- Déplacement de phase à 4 états QPSK (*Quadrature Phase-Shift Keying*)
- Modulation d'amplitude en quadrature à 16 états 16-QAM (*16-Quadrature Amplitude Modulation*)
- Modulation d'amplitude en quadrature à 64 états 64-QAM (*64-Quadrature Amplitude Modulation*)

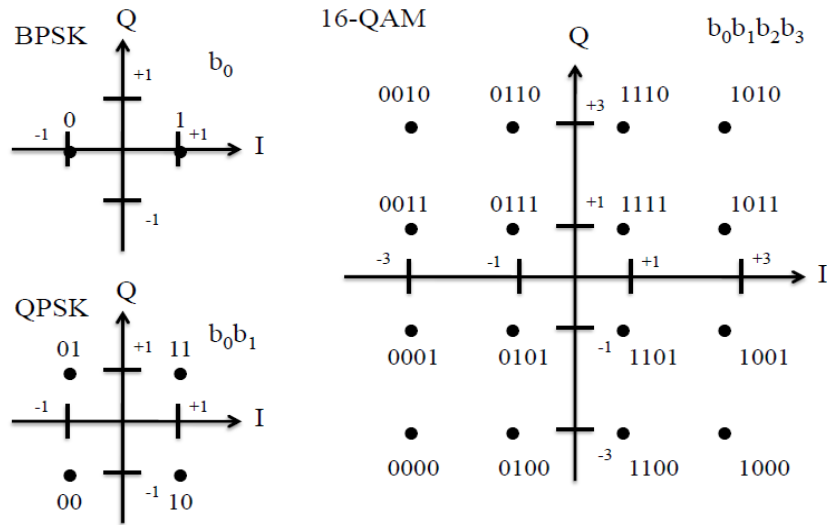


FIGURE 1.1 – Différents types de *mapping*

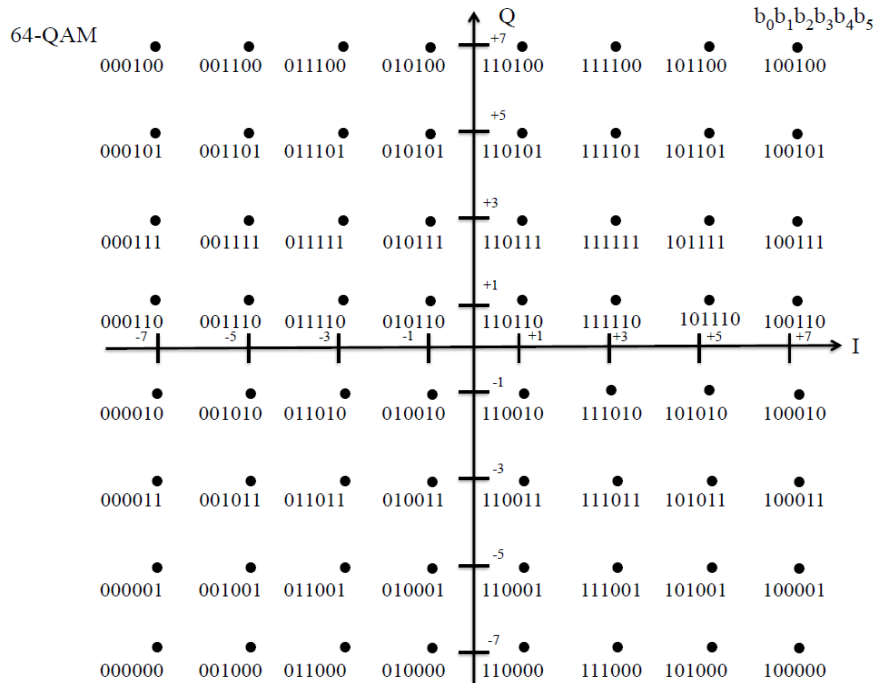


FIGURE 1.2 – Modulation QAM à 64 états

Les figures 1.1 et 1.2 représentent quelques différents types de modulation ainsi que les correspondances avec les expressions binaires utilisées. Un type de codage binaire, couramment utilisé, est le *mapping* de Gray qui permet de ne modifier qu'un seul bit entre deux symboles voisins dans la constellation.

Du côté récepteur, l'opération duale permettant d'extraire l'information binaire du symbole complexe s'appelle le *demapping*. A noter que le demappeur peut fournir des sorties souples qui constituent une estimation pondérée de la séquence de bits émis [21].

1.3.2 Canal de transmission

Comme indiqué dans la section précédente, un canal de transmission est un support de communication d'information. Il relie la source au destinataire et permet l'acheminement du message. En théorie de l'information, on utilise généralement un canal de communication dit discret. Celui-ci ne transmet que des nombres binaires et, le plus souvent, seulement 0 ou 1, c'est-à-dire des bits d'information. Le système de transmission est représenté en utilisant un modèle de temps discret. Les informations émises et les données reçues forment alors des échantillons indexés par le temps. Dans la suite, différents modèles du canal de transmission sont envisagés.

1.3.2.1 Modèle d'un canal BABG

Le canal à Bruit Additif Blanc Gaussien (BABG), appelé aussi canal AWGN (*Additive White Gaussian Noise*) est un canal à entrée discrète et à sortie continue. Dans ce type de canal, les informations transmises sur le canal sont perturbées par un bruit additif [22] lors de la transmission. Le bruit Gaussien est un bruit qui suit une loi normale de moyenne et de variance données. Ce bruit blanc est modélisé par une variable aléatoire gaussienne de moyenne nulle. Pour le canal BABG, le signal reçu y_k s'expriment en fonction du signal émis x_k par l'équation suivante :

$$y_k = x_k + b_k \quad (1.1)$$

où b_k est un bruit blanc Gaussien de moyenne nulle indépendant du signal x_k et de variance σ^2 . La densité de probabilité de transition du canal BABG est de la forme suivante :

$$p(y_k/x_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_k - x_k)^2}{2\sigma^2}\right) \quad (1.2)$$

Le canal BABG peut être utilisé pour modéliser une transmission sans interférence entre symboles perturbée par un bruit blanc centré et Gaussien.

1.3.2.2 Modèle d'un canal Rayleigh

Dans le cas d'un canal Rayleigh [23], les données transmises subissent une atténuation qui peut varier dans le temps. Cette atténuation peut être différente d'un symbole à l'autre. Elle peut aussi rester constante lors de la transmission d'un bloc de données. La relation qui associe l'entrée x_k et la sortie y_k d'un canal Rayleigh s'écrit :

$$y_k = h_k \cdot x_k + b_k \quad (1.3)$$

b_k est un bruit BABG et les coefficients d'atténuation h_k suivent une loi de Rayleigh unitaire. La loi de Rayleigh est une loi de probabilité à densité. Elle apparaît comme la norme d'un vecteur Gaussien bi-dimensionnel dont les coordonnées sont indépendantes, centrées et de même variance. Cette loi de probabilité est baptisée d'après Lord Rayleigh en 1904. La densité de la loi de Rayleigh est donnée par :

$$p(x; \sigma) = \frac{x}{\sigma^2} \exp\left(\frac{-x^2}{2\sigma^2}\right) \quad (1.4)$$

La figure 1.3 illustre la densité de probabilité de la loi de Rayleigh, c'est la représentation la plus connue de cette loi. En effet, lors de la transmission des symboles, l'atténuation du canal peut être importante et dégrader la qualité de la transmission. On parle alors d'évanouissements du canal.

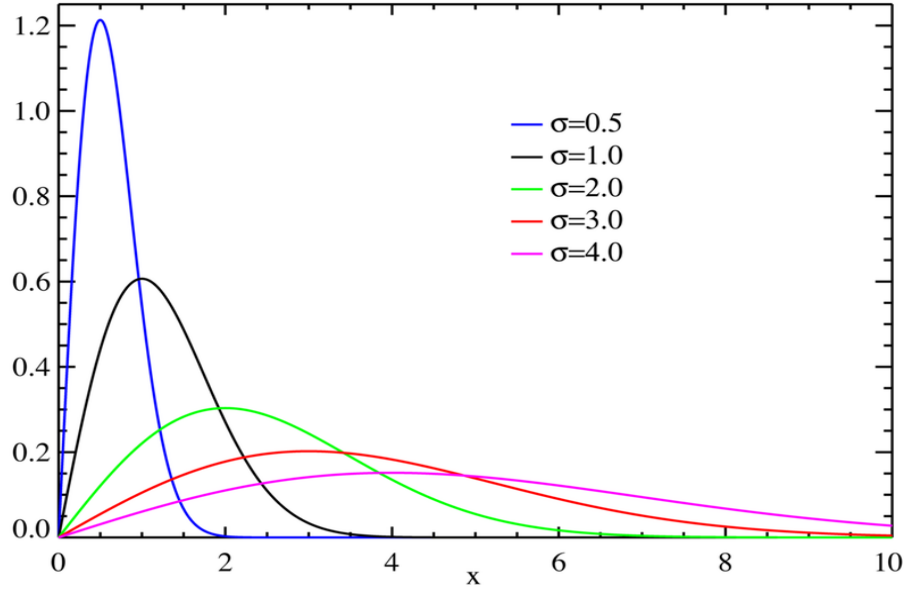


FIGURE 1.3 – Densité de probabilité de la loi de Rayleigh

1.4 Codage et décodage canal

Le but essentiel d'un système de communication est de transmettre des informations entre deux points avec le maximum d'efficacité et de fiabilité. Claude Shannon a posé les bases de la théorie de l'information en 1948 [24]. La théorie de l'information permet d'évaluer quantitativement le contenu d'un signal porteur d'un message et de déterminer la capacité d'un système de communication à acheminer des informations. Un code correcteur d'erreurs permet d'améliorer la qualité de transmission sur un canal.

D'une manière générale, le principe des codes correcteurs d'erreurs est d'ajouter de la redondance au message à coder, dans le but d'utiliser cette information pour la détection voire la correction d'erreurs. Le décodeur de canal exploite la redondance introduite par le codeur de canal pour essayer de détecter puis, éventuellement de corriger les erreurs introduites lors de la transmission. Dans le cas binaire dans l'alphabet $\{0,1\}$ noté F_2 , à partir d'un message de K éléments binaires d'information issus de la source, le codeur génère un mot de code de N éléments binaires.

Le rendement du code noté $R = K/N$ indique la proportion de redondance introduite par le codage. La distance de Hamming [25] entre deux mots de code est définie par le nombre de bits qui y diffèrent. La plus petite distance de Hamming pour l'ensemble des couples de mots distincts d'un code est appelée distance minimale du code notée d_{min} . La capacité de correction d'un code est obtenue par la distance minimale d_{min} du code. Un codage canal de distance minimale d_{min} peut corriger jusqu'à t erreurs si et seulement si :

$$d_{min} \geq (2t + 1)$$

La recherche de l'élément le plus proche dans un ensemble de grande dimension est un problème algorithmique qui se résoud par une recherche séquentielle exhaustive.

1.4.1 Les codes correcteurs d'erreurs

Deux grandes familles de code ont été développées et sont actuellement utilisées. Ils sont toujours l'objet de travaux de recherche. La première est celle des codes en blocs, tels que les codes BCH (Bose, Chaudhuri et Hocquenghem) [26] ou de Reed-Solomon [27]. Le message à transmettre est décomposé en blocs codés indépendamment. Le décodage se fait également par blocs, avec des recherches dans des espaces réduits.

La seconde famille, les codes convolutifs, forment une classe extrêmement souple et efficace de codes correcteurs d'erreurs. Pour ces codes, chaque bloc de n éléments binaires en sortie du codeur dépend non seulement du bloc de k éléments binaires présent à son entrée mais aussi de m blocs présents précédemment. Les codes convolutifs introduisent par conséquent un effet mémoire d'ordre m . Donc, le codage se fait grâce à des registres à décalages, alimentés par les bits du message. Comme pour les codes en blocs, le décodage d'un code convolutif va consister à rechercher la séquence la plus proche de la séquence reçue. Si le nombre de séquences possibles étant généralement très important, le décodage est donc d'une complexité prohibitive.

Des algorithmes permettant de contourner cette difficulté ont été développés. Parmi ces algorithmes on peut citer l'algorithme de Viterbi [28] qui utilise le graphe de treillis pour rechercher la séquence la plus proche. Cet algorithme est beaucoup utilisé et surtout bien adapté pour le décodage des codes de longueur peu élevée. Une percée majeure dans le domaine des communications numériques a été faite en 1993 avec l'introduction des turbo-codes [3], [29]. Cette technique est applicable aux deux types de codes en blocs et convolutifs, utilise deux codeurs différents. Le résultat du décodage de l'un sert d' *a priori* à l'autre de façon itérative.

Par ailleurs, les codes LDPC (Low Density Parity Check) redécouverts en 1996 [30] permettent également d'atteindre d'excellentes performances. Ces codes forment une famille de codes en blocs linéaires. Ils se caractérisent par une matrice creuse de contrôle à vérification de parité. Les codes LDPC ont été définis pour la première fois par Gallager en 1962 [8]. Mais ils ont eu peu de succès à l'époque en raison de leur complexité importante et de la découverte des codes de Reed-Solomon. Ces travaux ont été oubliés pendant 30 ans mais après l'invention des turbo codes, les codes LDPC ont été redécouverts à la fin des années 1990 [30]. En effet, les progrès des techniques de décodage ont permis aux codes LDPC d'être exploités dans les applications pratiques.

1.4.2 Les codes LDPC binaires

1.4.2.1 L'équation de parité

Les codes LDPC sont des codes construits à partir d'un code de parité. Une équation de parité représentée par la Figure 1.4, est une équation qui relie, par l'opérateur *ou exclusif* noté \oplus , des données binaires. Cette équation vérifie si le nombre total de 1 dans l'équation est pair ou nul. Dans la figure 1.4, les rectangles représentent les données binaires b_i , aussi appelées nœuds de variable. Le cercle contenant l'opérateur \oplus représente l'équation de parité, appelée aussi nœud de parité.

L'observation du canal i_j est appelée information intrinsèque de la valeur b_j . Nous considérons que les variables binaires b_0, b_1, b_2 sont liées par la contrainte de parité $b_0 \oplus b_1 \oplus b_2 = 0$. Elles forment le mot de code (b_0, b_1, b_2) .

Les Logarithmes de Rapport de Vraisemblance (LRV) : $i_1 = LRV(b_1)$ et $i_2 = LRV(b_2)$ sont supposés connues. Le LRV est une manière pratique de représenter l'information pondérée d'une valeur binaire [31]. Il est à noter également qu'un $LRV(b_j)$ d'un bit $b_j \in \{0, 1\}$ est défini par :

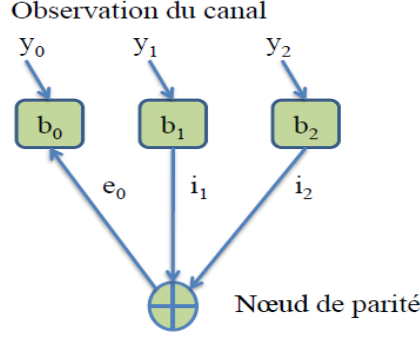


FIGURE 1.4 – Représentation graphique d’une équation de parité binaire pour trois bits.

$$LRV(b_j) = \ln \left(\frac{Pr(b_j = 0)}{Pr(b_j = 1)} \right) \quad (1.5)$$

Dans l’exemple de la figure 1.4, la contrainte de parité permet de déduire que deux mots de code permettent de retrouver le bit b_0 :

- Les mots de code (0,0,0) et (0,1,1) pour lesquels $b_0 = 0$

$$Pr(b_0 = 0) = Pr(b_1 = 0) \times Pr(b_2 = 1) + Pr(b_1 = 1) \times Pr(b_2 = 0)$$

- Les mots de code (1,0,1) et (1,1,0) pour lesquels $b_0 = 1$

$$Pr(b_0 = 1) = Pr(b_1 = 0) \times Pr(b_2 = 0) + Pr(b_1 = 1) \times Pr(b_2 = 1)$$

L’observation suivante est ainsi obtenue :

$$e_0 = LRV(b_0) = \ln \left(\frac{Pr(b_j = 0)}{Pr(b_j = 1)} \right) \quad (1.6)$$

Cette observation est appelée information extrinsèque de la valeur binaire b_0 à partir des informations intrinsèques provenant des valeurs binaires b_1 et b_2 . Cette observation est indépendante de i_0 . La valeur finale est obtenue en appliquant l’opération d’addition sur i_0 et e_0 . Nous notons que ce processus est le même pour tous les symboles d’observations du canal.

1.4.2.2 Représentation des codes LDPC binaires

Les codes LDPC appartiennent à la catégorie des codes correcteurs d’erreurs linéaires en bloc. La particularité de ce type de code est que sa matrice de parité H est très creuse, c’est-à-dire que le rapport entre le nombre de "1" et de "0" dans la matrice de parité tend vers 0. De ce fait, les processus de décodage peut être effectué plus rapidement si l’on exploite les propriétés de la matrice creuse. Cette matrice est de taille $M \times N$, où M est le nombre de bits de redondances et N est le nombre de bits du mot de code.

Cette matrice peut être vue comme un système linéaire de M équations de parité. Un mot de code défini par la matrice H est un vecteur binaire dont les N bits vérifient simultanément les M équations de parité. Ce système d’équations linéaires peut être représenté par un graphe de Tanner [32] ou un graphe bipartite. Dans ce graphe, des branches sont reliées à des nœuds appartenant à

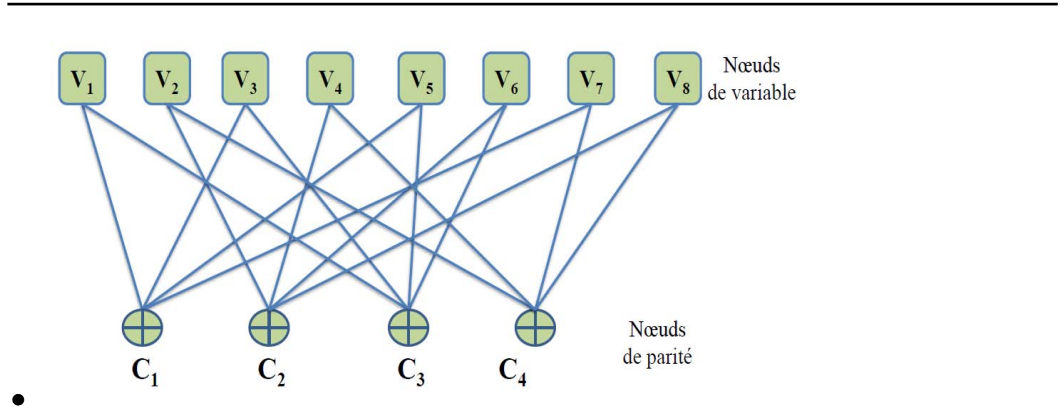


FIGURE 1.5 – Représentation de type graphe bipartite pour un code LDPC binaire.

deux classes différentes :

Les nœuds de variable, correspondent aux bits des mots de code $(V_j, j \in (1, \dots, n))$, et donc aux colonnes de H .

Les nœuds de parité, correspondent aux équations de parité $(C_p, p \in (1, \dots, m))$, et donc aux lignes de H .

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Par exemple, la matrice de parité H (au dessus) de taille $N = 8$ et $M = 4$, peut être représentée par un graphe bipartite comme celui de la figure 1.5.

Les codes LDPC peuvent être de deux types : les codes réguliers et les codes irréguliers. Les codes LDPC réguliers sont les codes dont le nombre de 1 par ligne, noté d_v et le nombre de 1 par colonne noté d_c sont constants. Dans ce cas, nous disons que les variables sont de degré d_v , les parité de degré d_c . Un tel code est noté code LDPC (d_v, d_c) . Le code LDPC de l'exemple précédent illustré dans la figure 1.5 est un code LDPC régulier de $d_v = 2$ et $d_c = 4$.

Par extension, les codes LDPC irréguliers sont les codes définis par des matrices de contrôle de parité où le nombre de 1 par ligne et/ou par colonne n'est pas constant. Noter qu'un cycle dans un graphe bipartite est un chemin sur le graphe qui permet de partir d'un nœud et de revenir à ce même nœud sans passer par la même branche. La taille d'un cycle est donnée par le nombre de branches contenues dans le cycle.

Le graphe étant bipartite, la taille des cycles est paire. En anglais, la taille du cycle le plus court dans un graphe est appelée *girth* [32]. L'existence des cycles courts influe négativement sur les performances d'un code LDPC. En effet, dans un tel cas il y a un rebouclage de l'information ce qui va détruire l'hypothèse d'indépendance des informations extrinsèques et rend l'algorithme de décodage non convergent vers la bonne solution. Revenons à notre exemple qui contient un cycle de longueur 4 présenté par des connexions en tirets rouge dans la figure 1.6.

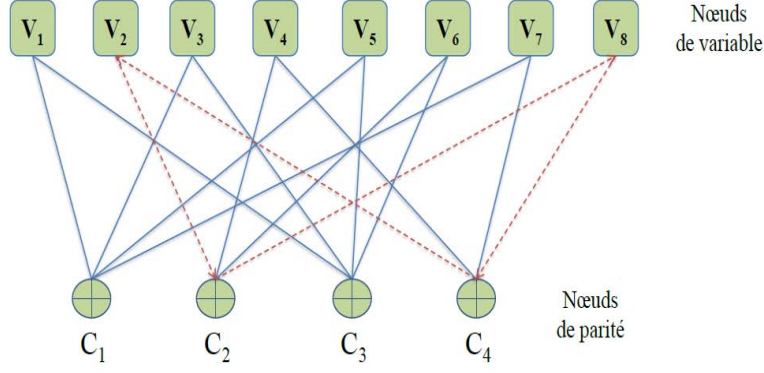


FIGURE 1.6 – Représentation d'un cycle de longueur 4.

1.4.2.3 Décodage itératif des codes LDPC binaires

Les algorithmes de décodage des codes LDPC sont des algorithmes itératifs utilisés le principe de passage de messages. On distingue dans ces algorithmes deux principales étapes. Premièrement les messages correspondant aux nœuds de variable sont transmis à l'ensemble des nœuds de parité pour la mise à jour des calculs de parités. Deuxièmement les messages de nœuds de parité sont transmis vers les nœuds de variable pour les calculs au sein des nœuds de variables. L'ensemble de ces deux étapes constitue une itération de l'algorithme de décodage. Le décodage est itératif. Il peut s'arrêter si le mot de code obtenu est valide ou après un nombre maximum d'itérations.

Le principe de ces algorithmes est que le message sortant sur la $j^{\text{ème}}$ branche est une fonction de tous les messages entrants sur ce nœud sauf celui de la $j^{\text{ème}}$ branche. Ainsi, les messages envoyés par un nœud de variable V à un nœud de parité C ne prends pas en compte le message échangé de V vers C lors de l'itération précédente. Par exemple, la figure 1.7.a illustre le calcul effectué au sein d'un nœud de variable. Le nœud de variable V_i est calculé en fonction de l'information intrinsèque observée du canal i_i et les messages extrinsèques $e_m, m = (1, \dots, 4)$, transmis par les autres nœuds de parité auxquels le nœud V_i est connecté. La valeur finale de V_i est obtenue par cette relation :

$$t = i_i + \sum_{m=1}^4 e_m \quad (1.7)$$

Dans la figure 1.7.b, nous illustrons le calcul des messages envoyés par un nœud de variable V_i aux nœuds de parité C_m sous la forme suivante :

$$a_i = i_i + \left(\sum_{m=1, m \neq i}^4 e_m \right) = (t - e_i) \quad (1.8)$$

La même stratégie est appliquée pour les nœuds de parité (figure 1.8). Donc, l'information extrinsèque $e_i, i = (1, \dots, 4)$, envoyée au nœud de variable V_i est calculée sous cette relation :

$$e_i = \bigoplus_{j \neq i} a_j = a_1 \oplus a_2 \dots a_{i-1} \oplus a_{i+1} \dots \oplus a_{d_c} \quad (1.9)$$

Gallager [8] puis Tanner [32] ont proposés un algorithme de décodage pour les codes LDPC basé sur ce principe. On le retrouve dans la littérature sous les appellations suivantes : algorithme par propagation de croyances (*Belief Propagation*, BP) ou algorithme Somme-Produit (*Sum-Product*, SP). Comme son nom l'indique, cet algorithme propage le long des branches du graphe associé au code des

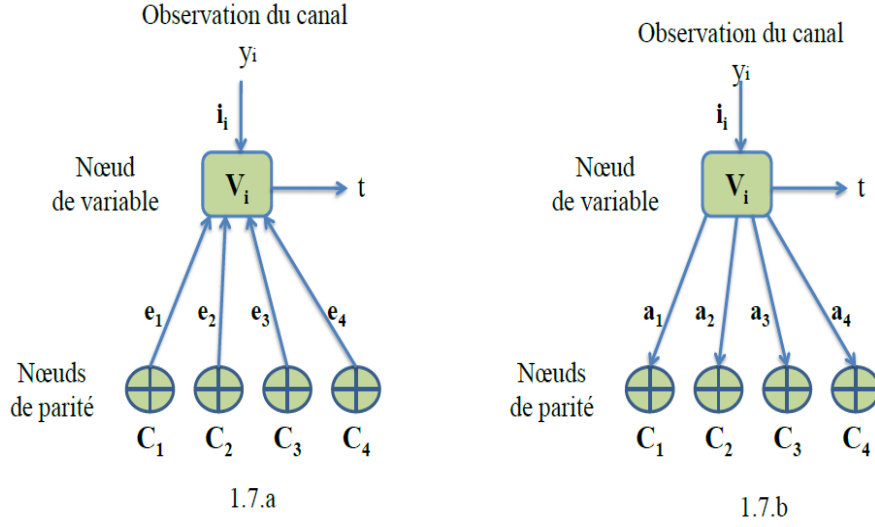


FIGURE 1.7 – Représentation de calcul d'un nœud de variable.

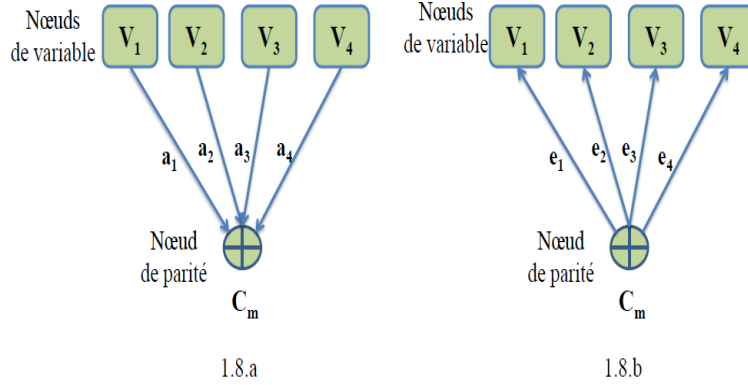


FIGURE 1.8 – Représentation de calcul d'un nœud de parité.

messages, qui sont des probabilités. A chaque branche sont associé deux messages, un dans chaque direction de propagation de l'information en appliquant le principe de passage des messages.

1.4.2.4 Algorithme de décodage Min-Somme

L'algorithme Min-Somme (*Min-Sum*, MS) [33] est la version modifiée de l'algorithme SP [34], où l'opération de vérification de nœud de parité est simplifiée pour réduire considérablement la complexité de l'algorithme. Dans l'algorithme MS, les messages intrinsèques et extrinsèques sont calculés dans le domaine logarithmique. En utilisant ce domaine, nous obtenons une mise à jour au sein d'un nœud de parité nécessite de ne réaliser que des additions simples au lieu d'avoir une mise à jour basée sur le calcul des sommes de produits. En effet, la complexité de la mise en œuvre matérielle augmente d'une façon linéaire par rapport à la taille des messages [35].

1.4.2.5 Décision dure et décision souple

Le décodage à décision dure consiste à prendre une décision sur chaque symbole observé en sortie du canal [36]. L'ensemble de ces décisions binaires est envoyé vers les nœuds de parité. Après que l'équation de parité soit vérifiée, le nœud renvoie vers chaque nœud de variable le symbole qui satisfait l'équation de parité. Le rôle du nœud de variable est alors de prendre une décision de décodage à

partir des éléments envoyés par les nœuds de parité et à partir du symbole issu de la décision initiale [30].

Cette décision est ensuite renvoyée vers le nœud de parité. Le décodage à décision souple repose sur le même principe que pour une décision dure sauf qu'il possède des messages contenant des valeurs réelles (probabilités, LRV : log rapport de vraisemblance) à la place des valeurs binaires.

Ainsi, à partir des observations du canal, les probabilités sont calculées a priori sur les valeurs de chaque symbole du mot de code. Dès lors, c'est la probabilité de chaque symbole, qui est exploitée dans le processus de décodage.

1.4.3 Les codes LDPC non-binaires

En 1998, Davey *et al.* [9] ont généralisé les codes LDPC binaires au cas non-binaire. Les codes LDPC non-binaires sont des codes basés sur une matrice de parité creuse dont ses éléments non-nuls sont définis sur le corps de Galois [37] d'ordre q . Les principales raisons d'employer des codes LDPC non-binaires sont liées aux points suivantes :

- Au cas d'un mots de code de taille finie, les algorithmes sous-optimaux des décodeurs comme l'algorithme BP et ces dérivés à complexité réduit, sont beaucoup plus proches de décodage optimal de Maximum de Vraisemblance MV que leurs homologues binaires [10].
- Il est bien connu que les inconvénients du décodage sous-optimal par passage de message viennent de la dépendance des messages qui transitent sur les branches de graphe du code induite par des structures topologique spécifique comme : les cycles, *stopping* ou *trapping set*. Le mauvais comportement du décodeur BP sur ces sturctures est accentué si les messages qui initialisent le décodeur sont déjà corrélés par le canal. Généralement, l'utilisation des codes LDPC non-binaires aide à minimiser les effets de corrélation des messages.
- Les travaux de recherche présentés par [10], [38], [39] montrent que la matrice de parité d'un code LDPC non-binaire est plus creuse que celle de son homologue binaire pour un rendement et une taille du mot de code identique. Par conséquent, le cycle d'un code LDPC non-binaire est généralement plus long que celle d'un code binaire. Ainsi, la répartition au sein d'un graphe LDPC-NB aide à éviter les cycles courts (*girth*) et réduit également l'effet d'arrêter les échanges des messages ou *trapping set* [38].
- Dans le cas d'une transmission d'un mot de code LDPC binaire avec une modulation M-QAM d'ordre élevé ($M \geq 64$), le demapper Maximum *A Posteriori* MAP fournit à la réception des fiabilités corrélées au niveau binaire, ce qui implique un mauvais de comportement du décodeur BP. Cependant, si le code LDPC est construit dans un corps d'ordre égal ou plus haut que l'ordre de modulation, le décodeur non-binaire est initialisé avec des messages non-corrélés ce qui aide le décodage BP à se rapprocher du décodage optimal MV.

Par conséquent, les codes LDPC non-binaires peuvent être facilement combinés avec des mappings d'ordre élevés et avec des systèmes à multiple antennes, qui sont capables de supporter des transmis-

sions de données à haut débit [40] en atteignant un gain significatif en terme de correction d'erreurs [41]. Leurs performances et les progrès en terme de capacité de calcul les ont remis à l'ordre du jour [42], [43], [44], [45]. Comme dans le cas binaire les codes LDPC non-binaires peuvent être représentés sous deux formes. La première les décrit par des matrices, comme tous les codes de blocs linéaires. La seconde repose sur une approche graphique ou graphe bipartite [32].

1.4.3.1 Représentation matricielle

Les codes LDPC non-binaires sont des codes en blocs linéaires construits sur des corps de Galois non-binaires [37] à q éléments où $GF(q) = \{\alpha_0, \dots, \alpha_{q-1}\}$ avec α_i sont les éléments de ce corps. Une représentation de ce corps est obtenue en considérant les polynômes de degré inférieur ou égal à $q - 1$ et à coefficients dans le corps fini $GF(2) = \{0, 1\}$ muni de l'addition modulo 2 [37].

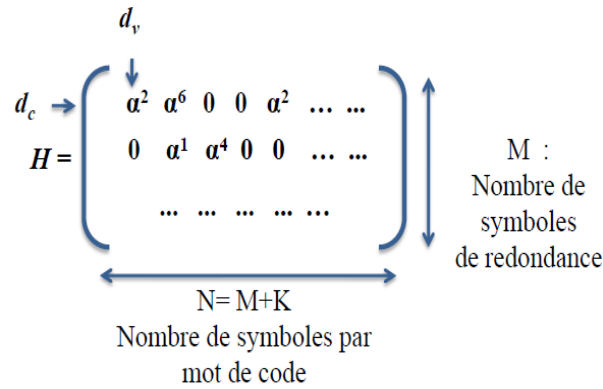


FIGURE 1.9 – Représentation matricielle d'un nœud de parité à contrainte non-binaire.

Un code LDPC non-binaire dans $GF(q)$ est défini par une matrice creuse de parité H de dimension $(N - K) \times N$, où K représente le nombre de symboles d'information et N représente le nombre des symboles codés. Les éléments non nuls appartiennent à $GF(q)$. Les mots de code V_i , $i = 1, \dots, N$ du code défini par la matrice H sont les mots dont les N symboles vérifient simultanément les $(N - K)$ équations de parité.

Une représentation matricielle est donnée dans la Figure 1.9. Gallager a défini une matrice de parité (N, d_v, d_c) par N colonnes contenant d_v (resp. d_c) éléments non nuls dans chaque colonne (resp. ligne), et chaque matrice a $m = N \times (d_v/d_c)$ lignes et son rendement est $R = 1 - (d_v/d_c)$. H est la matrice de parité telle :

- α_k sont les éléments du corps de Galois \in à $GF(q)$, d'où $k = \{0, \dots, q-1\}$.
- N est le nombre de colonnes égal à la longueur du mot de code.
- $M = N - K$ est le nombre de lignes égal au nombre de symboles de redondance.

1.4.3.2 Représentation graphique sous forme de graphe bipartite

La représentation donnée dans la figure 1.10 est un graphe bipartite ou graphe de Tanner associé au code LDPC non-binaire (N, M) . Dans un graphe bipartite non-binaire, les branches sont liées à trois nœuds différents :

- **Nœuds de variable** $\{V_i, i = 1, 2, \dots, N\}$, correspondent aux symboles non-binaires par mots de code.

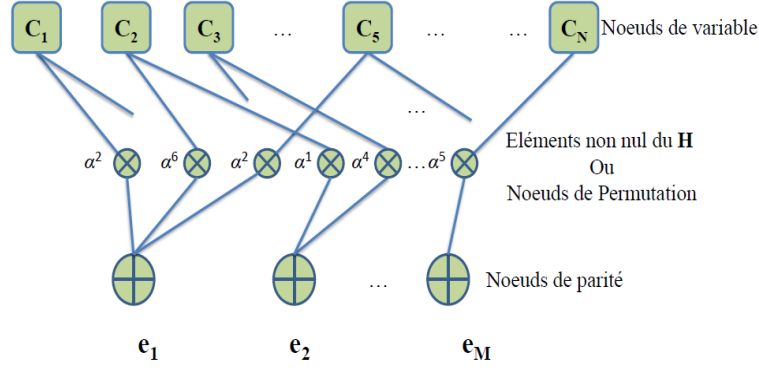


FIGURE 1.10 – Représentation graphique associé au code LDPC non-binaire.

- **Noeuds de parité** $\{C_m, m = 1, 2, \dots, M\}$, correspondent aux équations de parité défini par la matrice H .
- **Noeuds de permutation** $\{\alpha^q \in GF(q)\}$, correspondent aux éléments non nuls de la matrice H .

Pour chaque élément non nul α_k de la matrice de parité, nous avons une branche dans le graphe. Si l'élément α_k de la $m^{\text{ème}}$ ligne et de la $i^{\text{ème}}$ colonne est non nul, nous pouvons connecter le $m^{\text{ème}}$ noeud de parité au $i^{\text{ème}}$ noeud de variable en multipliant ce dernier par α_k . Par exemple illustré à la figure 1.10, nous connectons le 5^{ème} noeud de variable au 1^{er} noeud de parité en passant par le noeud de permutation ou l'élément de la matrice de parité α_2 .

1.4.3.3 Principe de codage pour un code LDPC non-binaire

La méthode de codage la plus traditionnelle des codes LDPC non-binaires se base sur la déduction de la matrice génératrice G à partir de la matrice de contrôle H définie sous une forme systématique. On note que lorsque les mots du code sont constitués de la concaténation des données et de la redondance, le code est dit alors systématique.

Cependant, la matrice de contrôle de parité H des codes LDPC n'est pas sous une forme systématique donc on utilise l'élimination gaussienne [46] pour transformer H sous cette forme :

$$H = [P_{k,(N-K)} | I_{(N-K)}] \quad (1.10)$$

où I_{N-K} est la matrice d'identité de taille $(N - K)^2$. Dans ce cas, la matrice génératrice est sous la forme suivante :

$$G = [I_k | P^T] \quad (1.11)$$

L'inconvénient est que cette matrice n'est pas nécessairement de faible densité. Une première solution est de mettre H sous une forme presque triangulaire inférieure [47]. L'autre solution est de construire H sous une forme cyclique permettant d'obtenir un codage plus facile [47].

Soit E l'ensemble des messages d'information de taille K qu'on désire coder. Si il existe une application bijective $\phi : E \rightarrow GF(q)^N$, telle que $C = \text{Im}(\phi) \in GF(q)^N$ où C est un code en bloc de taille N , l'application ϕ constitue un codeur pour le code C .

Ainsi, C constitue un sous espace linéaire du corps $GF(q)^N$. On appelle matrice génératrice d'un code linéaire une matrice G de dimension $K \times N$ dont les lignes forment une base du code. On définit les vecteurs de base $[g_0, \dots, g_{k-1}]$ tels que le mot de code c peut être mis sous la forme :

$$c = (u_0 \times g_0) + \dots + (u_{K-1} \times g_{K-1}) \quad (1.12)$$

avec $u_i \in GF(q)$, $i = 0, \dots, K-1$, les symboles d'information correspondant au mot de code c . L'expression matricielle de l'équation (1.13) est donnée par :

$$c = u.G \quad (1.13)$$

avec u le vecteur ligne $[u_0, \dots, u_{K-1}]$ et G la matrice génératrice de taille $K \times N$ du code C .

1.4.3.4 Décodage des codes LDPC non-binaires

Les algorithmes de décodage, précédemment cités, utilisés pour les codes LDPC binaires peuvent être étendus aux codes LDPC non-binaires. Les algorithmes de décodage des codes LDPC non-binaires sont des algorithmes itératifs basé sur le principe de propagation de croyance. Ces algorithmes ont l'avantage d'offrir une large diversité en terme de possibilités d'implantation. Ils peuvent être implantés dans le domaine des probabilités [8], [9] connus sous l'appellation *Sum-Product*, ou dans le domaine logarithmique [44], [45] sous le nom *Min-Sum*.

L'algorithme *Min-Sum* est la version modifiée de l'algorithme *Sum-Product*. Les principes généraux de fonctionnement des algorithmes de décodage des codes LDPC dans les corps de Galois $GF(q)$ s'appliquent de la même manière que dans le cas binaire. Les symboles du code non-binaire étant considérés comme des variables aléatoires dans $GF(q)$ où q est l'ordre du corps. Chaque nœud de variable envoie aux nœuds de parité auxquels il est associé un message de q valeurs sur l'estimation du symbole non-binaire (information *a priori*). L'ensemble des messages *a priori* reçus permet à la contrainte de parité de calculer, puis de retourner les q informations extrinsèques. Le traitement successif des nœuds de variable puis des nœuds de parité constitue une itération. Au cours de chaque itération, il y a un échange bilatéral de messages entre les nœuds de parité et les nœuds de variable.

L'algorithme de propagation de croyance comprend donc deux étapes de calcul. Premièrement, le traitement dans les nœuds de données est une multiplication termes à termes des $d_v - 1$ messages entrants. Puis, le traitement dans les nœuds de parité consiste en une marginalisation des d_c messages entrant conditionnellement à l'équation de vérification de parité :

$$\sum_{t=1}^{d_c} h_t \otimes c_t = 0 \quad (1.14)$$

où h_t sont les éléments non-nuls de la matrice H et c_t les symboles correspondants du mot de code.

Cette seconde étape est très complexe et représente l'obstacle majeur au décodage de codes LDPC non-binaires dans des corps de Galois d'ordre élevés. La complexité par nœud de parité a pour ordre de complexité $O(q^2)$. Malheureusement, sous cette forme l'algorithme de propagation de croyance requiert des multiplications et des divisions, ce qui est dommageable à toute implantation pratique.

En vue d'une implantation matérielle des décodeurs de codes LDPC non-binaires, il est nécessaire de proposer un décodeur utilisant des logarithmes de rapports de vraisemblance pour les messages (algorithme *Min-Sum*). En effet, ce dernier nécessite de ne réaliser que des additions. Par la suite, nous allons utiliser pour désigner le vecteur LRV d'une variable aléatoire $S \in GF(q)$, la notation suivante :

$$LRV(S) = [LRV(S[0]), LRV(S[1]), \dots, LRV(S[q-1])]^T \quad (1.15)$$

où

$$LRV(S[k]) = \log \left(\frac{P(S = \alpha_k)}{P(S = \alpha_0)} \right) \quad (1.16)$$

avec $P(S = \alpha_k)$ la probabilité que la variable aléatoire S est égale à $\alpha_k \in GF(q)$. Avec cette définition $LRV(S[0]) = 0, LRV(S[k]) \in R$.

1.4.3.5 Algorithme EMS (*Extended Min Sum*)

Les codes LDPC construits dans des corps de Galois non-binaires $GF(q)$ et décodés en utilisant l'algorithme de propagation de croyances (BP), représentent une solution efficace pour la correction d'erreurs de transmission dans le cas de trames courtes et de modulations d'ordres élevés. Les décodeurs associés ont l'inconvénient d'avoir une complexité en $O(q^2)$. Par ailleurs, l'espace mémoire nécessaire au stockage des messages a une complexité en $O(q)$.

En conséquence, les codes LDPC de corps d'ordre élevés ($q > 16$) ne peuvent être considérés en raison de la complexité de l'étape de décodage. Un algorithme de décodage simplifié pour les codes non-binaires a été présenté en 2005 [48] ayant une complexité dominée par $O(n_m^2)$, basé sur l'utilisation des $n_m \ll q$ valeurs les plus fiables lors de la mise à jour des messages par la fonction de parité. La structure de cette proposition consiste à modifier des messages permettant de réduire la complexité de stockage en $O(n_m)$ avec $n_m \ll q$, à l'aide d'une opération de troncature.

L'algorithme de décodage, appelé EMS *Extended Min Sum* [48],[49], a une complexité relativement faible. Cet algorithme est une simplification de l'algorithme *Min-Sum* dans le domaine logarithmique. Il s'agit d'une solution de décodeur non-binaire pratique pouvant concurrencer les décodeurs binaires. Nous utilisons une représentation logarithmique LRV des messages équivalente à l'équation (1.17). Un message $U_{vp} = \{U_{vp}[k]\}_{k=0,1,..,n_m-1}$ est le message allant du $i^{\text{ème}}$ nœud de variable au $j^{\text{ème}}$ nœud de permutation et $V_{pv} = \{V_{pv}[k]\}_{k=0,1,..,n_m-1}$ est le message provenant du $j^{\text{ème}}$ nœud de permutation vers le $i^{\text{ème}}$ nœud de variable. A l'aide des notations de la figure 1.11, on illustre les 3 étapes d'une itération de décodage de l'algorithme EMS :

Mise à jour des nœuds variable de degré d_v :

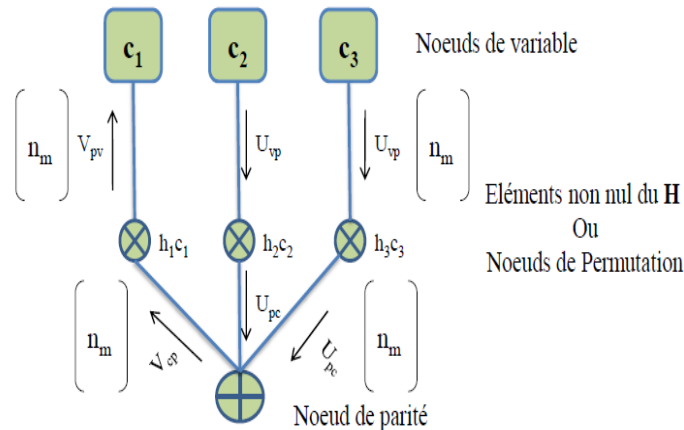


FIGURE 1.11 – Représentation graphique associé au algorithme EMS.

$$U_{vp}[k] = L[k] + \sum_{g=1, g \neq p}^{d_v} V_{pv}[k] - offset \quad (1.17)$$

où $k = 0, 1, n_m - 1$ et L représente les LRV des symboles provenant du canal. La méthode consistant à introduire un facteur de correction (offset) est régulièrement utilisée pour compenser la perte due à l'utilisation d'algorithmes sous-optimaux [48].

Etape de permutation des messages :

Cette étape est une conséquence directe de l'équation de parité non-binaire. En effet, l'algorithme de propagation de croyance employé pour la mise à jour des nœuds de parité dans le cas non-binaire utilise le produit entre les symboles du mot de code et leur valeur non-binaire h_t correspondante dans la matrice de parité. Cette multiplication se traduit par une permutation des LRV des symboles :

$$U_{pc}[k] = [U_{vp}[k] \otimes h_t] \quad (1.18)$$

avec $k = 0, 1, ..n_m - 1$. La transformation du graphe bipartite de la figure 1.11 permet l'écriture du processus de mise à jour de nœud de parité sous la forme d'un produit de convolution, similaire au cas binaire. La transformation en sens inverse de V_{cp} vers V_{pv} est faite à l'aide des valeurs h_t^{-1} .

Mise à jour des nœuds parité de degré d_c :

Le principe de l'algorithme EMS est de n'utiliser que n_m valeurs LRV en vérifiant l'équation de parité, afin de construire une fonction de fiabilité sous-optimale. Celle-ci est utilisée pour la mise à jour des nœud de parité, elle doit être très peu coûteuse en nombre d'opérations et dépend de l'équation de vérification de parité. L'algorithme EMS peut être implanté d'une manière efficace via la méthode récursive proposée dans [9] pour la mise à jour des nœuds de parité.

Cette méthode s'interprète comme une implantation type *{Forward/Backward}* du calcul des fiabilités. A chaque étape de récursion, appelée étape élémentaire, on considère seulement deux messages entrants pour effectuer le calcul des fiabilités illustrés dans la figure 1.12. Cette approche pour présenter les détails de l'étape de mise à jour des nœud de parité est basée sur le principe de la méthode récursive, chaque processus intermédiaire de vérification de parité comportant deux entrées.

Ainsi deux types de messages sont considérés dans la description de l'algorithme. Les messages de type U sont des messages U_{pc} , venant du graphe, les messages I sont des messages intermédiaires nécessaires à l'implantation récursive. Conformément à la représentation en graphe bipartite de la méthode récursive [9], pour des valeurs de $d_c > 4$ on distingue trois types d'étapes élémentaires :

- 1-) Les deux messages entrants sont de type U : étape U-U.
- 2-) Un des messages entrant est de type U et l'autre de type I : étape U-I.
- 3-) Les deux messages entrants sont de type I : étape I-I.

Soient $\{x_t^k\}_{t=(1,2)}$, $k = 0, 1, ..n_m - 1$, les messages entrant dans un nœud de parité élémentaire, c'est-à-dire une des étapes décrites ci dessus (U-U), (U-I) ou (I-I), et Y_3 le message sortant. On

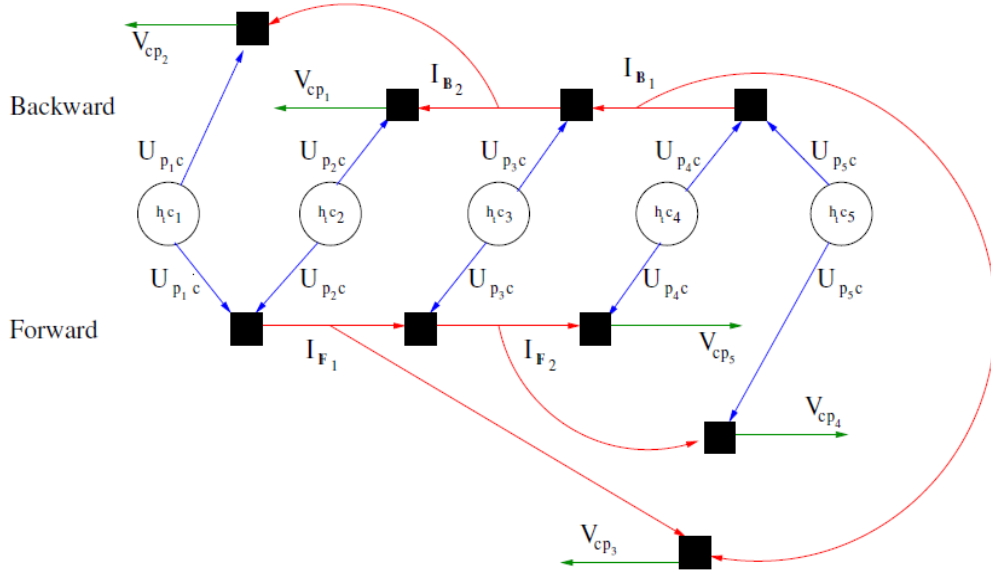


FIGURE 1.12 – Représentation graphique de la méthode récursive d'une équation de parité.

note de plus α_t^k les valeurs des éléments de $GF(q)$ correspondant à ces n_m valeurs. A l'aide de ces valeurs α_t^k et des contraintes de parité du codes LDPC, on construit l'ensemble de configurations :

$$S(\alpha_3^{(i)}) = \{(x_1^{(k)}, x_2^{(k)}) : \alpha_1^{(k)} + \alpha_2^{(k)} + \alpha_3^{(i)} = 0\} \quad (1.19)$$

$\forall k = 0, 1, \dots, n_m - 1$. A chacune des configurations, on associe une fiabilité définie comme suit :

$$LRV(\alpha_3^{(i)}) = \sum_{t=1}^2 (x_t^{(k)}) \quad (1.20)$$

où $i = 0, 1, \dots, n_m - 1$. Les valeurs $LRV(\alpha_3^{(i)})$ expriment la fiabilité, que l'équation de parité soit satisfaite lorsque $h_3 \otimes c_3$ est supposé égal à $\alpha_3^{(i)}$. Dans ces conditions, l'équation de mise à jour correspondant à une étape élémentaire est :

$$Y_3[\alpha_3^{(i)}] = \max_{[S(\alpha_3^{(i)}) \cup S(\alpha_0^{(i)})]} LRV(\alpha_3^{(i)}) \quad (1.21)$$

où $i = 0, 1, \dots, n_m - 1$, la fonction maximum servant à choisir la configuration la plus fiable.

1.5 Les systèmes MIMO

Les systèmes multi-antennes à l'émission et à la réception MIMO (Multiple Input Multiple Output) permettent d'accroître la capacité de transmission par rapport aux systèmes composés d'une seule antenne. C'est une technique, utilisée récemment dans les réseaux sans-fil et les réseaux mobiles (*Smart Phone* et tablette tactile), permettant des transferts de données à plus longue portée et à plus grande vitesse.

Cette technique de communication sans-fil utilisant des antennes multiples a été introduite en 1984 par Jack Winters [50]. En 1996, Foschini *et al.* ont inventé de nouvelles approches accroissant

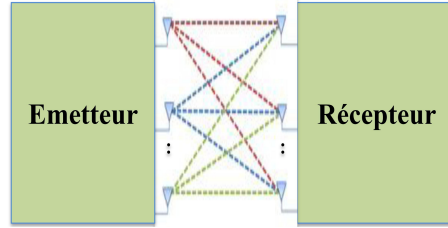


FIGURE 1.13 – Représentation d'un système de transmission à un canal MIMO.

l'efficacité du MIMO [5]. Puis, Foshini *et al.* [2] et Telatar *et al.* [1] ont démontré que la capacité théorique d'un canal MIMO avec N_t antennes à l'émission et N_r antennes à la réception croît linéairement avec le minimum des nombres N_t et N_r .

Le principe de base des systèmes MIMO consiste à combiner les signaux tant à l'émission qu'à la réception pour exploiter la diversité spatiale et donc réduire les effets des évanouissements ou pour augmenter le débit de transmission. Ils présentent deux avantages majeurs par rapport aux systèmes SISO. Le premier avantage est que cela permet d'améliorer la qualité du lien en s'affranchissant des évanouissements des canaux grâce à l'apport de la diversité spatiale. Le second avantage est d'augmenter le débit d'information sans augmenter la bande passante ou la puissance transmise.

Les systèmes MIMO sont l'un des principaux axes de développement pour augmenter les débits de transmission numérique. Nous assistons à un très rapide développement de cette technologie avec des applications déjà envisagées dans les réseaux locaux sans-fil et les réseaux de communication de 4ème génération. Par exemple, Les systèmes MIMO à 2 antennes à l'émission et 2 antennes à la réception (2x2) sont proposés pour le standard de 4ème génération et pour le standard de réseau local sans-fil IEEE 802.11n où l'objectif est d'atteindre des débits théoriques plus de 300 mégabits par seconde.

1.5.1 Chaîne de transmission numérique de type MIMO

Soit un système MIMO composé de N_t antennes à l'émission et de N_r antennes à la réception comme illustré sur la figure 1.14. Cette figure donne le schéma global de transmission dans lequel nous nous plaçons. Nous allons décrire les différents éléments de cette chaîne. Nous commençons par les caractéristiques des blocs à l'émission :

- **Codage canal** : Le codeur de canal reçoit les bits d'informations source et rajoute de la redondance afin de rendre les données robustes vis à vis des erreurs introduites par le canal. On note R , le rendement de codage entre le nombre de données d'informations transmises et le nombre de données totales transmises.
- **Modulation** : Les bits codés passent à l'étape de la modulation qui définit le processus par lequel le signal est transformé de sa forme originale en une forme adaptée au canal de transmission par un mapping de phase ou un mapping d'amplitude à plusieurs états.
- **Canal de transmission** : c'est le support utilisé pour véhiculer l'information. Le canal de transmission modélise la propagation d'un signal entre émetteur et récepteur. Le signal subit alors des perturbations. Ces perturbations incluent le bruit ambiant en réception.

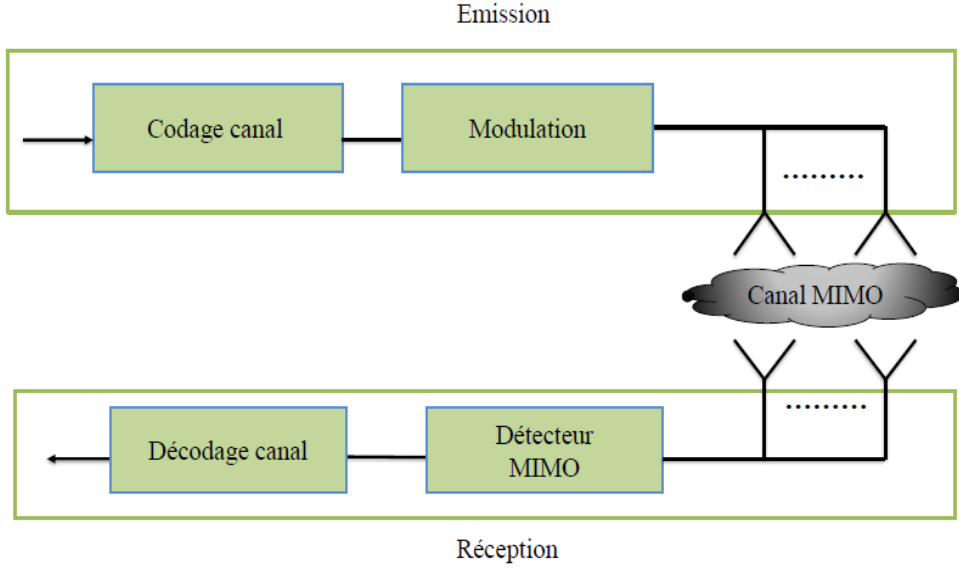


FIGURE 1.14 – Représentation d’une chaîne de transmission de type MIMO

Dans un canal MIMO, le signal reçu y_j à chaque instant sur la j^{ime} antenne de réception est la somme des symboles bruités issus des N_t signaux transmis :

$$y_j = \sum_{i=1}^{N_t} h_{ji}x_i + n_j \quad (1.22)$$

où h_{ij} est le gain du canal non sélectif en fréquence entre l’antenne d’émission i et l’antenne de réception j , x_i est le signal émis par la i^{me} antenne d’émission et n_j est le bruit additif qui est modélisé par la loi Gaussienne de variance $N_0/2$. Le rapport signal à bruit (RSB) par antenne de réception est défini par $RSB = E_s/N_0$. La matrice de canal MIMO notée H_c de dimension $N_r \times N_t$ est la suivante :

$$H_c = \begin{bmatrix} h_{11} & \cdots & h_{1N_t} \\ \vdots & \ddots & \vdots \\ h_{N_r1} & \cdots & h_{N_rN_t} \end{bmatrix} \quad (1.23)$$

En conséquence, le signal reçu peut s’écrire sous une forme matricielle :

$$\mathbf{y} = H_c \mathbf{x} + \mathbf{n} \quad (1.24)$$

où \mathbf{y} et \mathbf{n} sont respectivement les vecteurs de réception et de bruit de dimension $N_r \times 1$.

1.5.2 Capacité d’un système MIMO

La capacité d’un canal de transmission, notion introduite par Shannon [24] en 1948, mesure la quantité d’information maximum par unité de temps et par bande de fréquence pouvant être transmise à travers un canal particulier. Nous considérons un système MIMO avec N_t antennes d’émission et N_r antennes de réception. Supposons que le récepteur connaît parfaitement le canal sachant que l’émetteur ne dispose pas de cette connaissance du canal.

Nous rappelons que la capacité instantanée d’un canal SISO [2] non sélectif en fréquence s’écrit :

$$C(\rho) = \log_2(1 + \rho|h|^2) \quad (1.25)$$

où ρ est le rapport signal à bruit et h représente un évanouissement complexe du canal. La matrice H_c du canal MIMO peut se décomposer en plusieurs canaux SISO parallèles comme suit :

$$H_c = U \Sigma V^T \quad (1.26)$$

avec U et V des matrices unitaires et Σ la matrice diagonale contenant les racines carrées des valeurs propres de la matrice de corrélation $\mathbf{H}_c^T \mathbf{H}_c$. Ainsi, la capacité instantanée d'un canal MIMO peut être écrite comme dans [5] :

$$C(\rho) = \sum_{i=1}^r \log_2 \left(1 + \frac{\rho}{N_t} \lambda_i \right) \quad (1.27)$$

où λ_i est la valeur propre de la matrice de corrélation et r est le nombre de valeurs propres qui est égal à $\min(N_t, N_r)$. Cette équation peut également être décrite comme suit [1], [5] :

$$C(\rho, N_t, N_r) = \log_2 \det \left(\mathbf{I}_{N_r} + \frac{\rho}{N_t} \mathbf{H}_c \mathbf{H}_c^T \right) \quad (1.28)$$

où \mathbf{I}_{N_r} est une matrice unitaire de taille N_r . La capacité moyenne ou ergodique s'obtient en calculant l'espérance sur toutes les réalisations possibles du canal MIMO :

$$C(\rho, N_t, N_r) = E \left\{ \log_2 \det \left(\mathbf{I}_{N_r} + \frac{\rho}{N_t} \mathbf{H}_c \mathbf{H}_c^T \right) \right\} \quad (1.29)$$

Dans ce contexte, la capacité augmente linéairement avec la valeur minimale entre le nombre d'antennes à l'émission et celui à la réception [2]. En effet, l'utilisation de plusieurs antennes en émission et à la réception permet d'exploiter la dimension spatiale. L'exploitation de la diversité spatiale favorise l'amélioration des performances et du débit.

La figure 1.15 représente les capacités de différents systèmes MIMO pour canal de Rayleigh. La

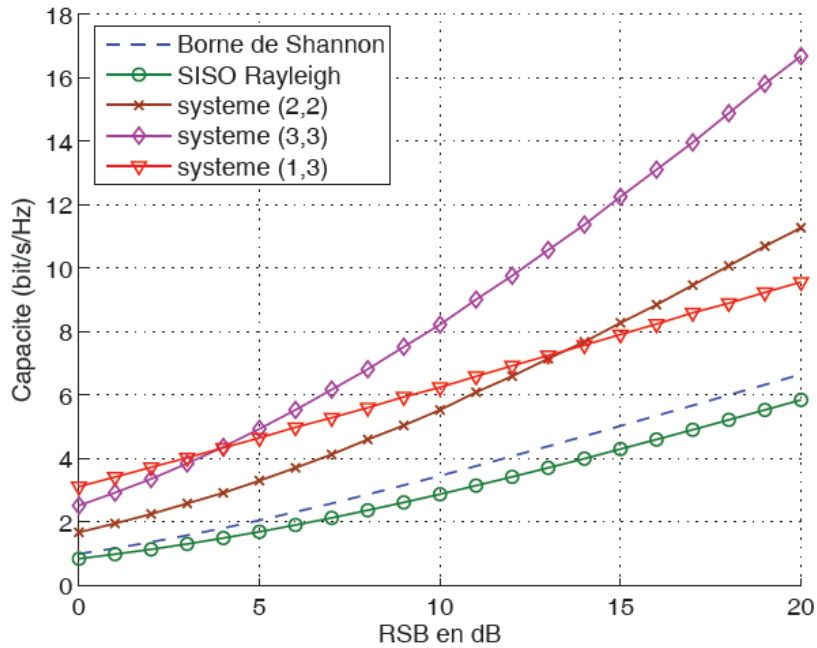


FIGURE 1.15 – Capacités de différents systèmes MIMO et capacité d'un système SISO par rapport à la borne théorique de Shannon dans le cas d'un canal de Rayleigh

capacité d'un système SISO et la borne théorique de Shannon sont aussi présentes. Nous montrons à l'aide de cette figure que les systèmes MIMO permettent d'atteindre voire de dépasser la limite de Shannon. De plus, les différents systèmes n'ont pas le même comportement en fonction du RSB. Par exemple, un système (1,3) est le meilleur à faible RSB mais n'est pas aussi efficace que les systèmes (2,2) et (3,3) lorsque le RSB devient important.

Gain de multiplexage pour des systèmes MIMO :

Un canal MIMO peut être interprété comme ensemble de canaux SISO en parallèle. Pour chacun des ces canaux, des informations différentes peuvent être transmises grâce aux canaux indépendants dont le nombre est égal à $\min(N_t, N_r)$. Ainsi, il est possible d'augmenter le débit d'information. Il en résulte un gain nommé gain de multiplexage.

Gain de diversité :

La diversité consiste à utiliser à la réception plusieurs répliques du signal afin de combattre les évanouissements. Un système MIMO employant N_t antennes à l'émission et N_r antennes à la réception peut atteindre un gain de diversité maximal de $d = N_t \times N_r$. Plus le nombre de répliques augmente, plus la probabilité que toutes les répliques subissent simultanément un évanouissement décroît exponentiellement en fonction de d [51]. Plusieurs techniques de transmission sont utilisées pour mettre en pratique les notions de diversité et de multiplexage. Nous présentons dans la suite de ce chapitre, quelques techniques de transmission des systèmes MIMO.

1.6 Techniques de transmission des systèmes MIMO

Le multiplexage consiste à faire passer plusieurs informations à travers un seul support de transmission. Il permet de partager une même ressource sur plusieurs antennes. En appliquant le principe de multiplexage, la trame à transmettre est divisée en sous-trames. Ces sous-trames sont ensuite transmises sur les différentes antennes suivant une répartition horizontale ou verticale. Foschini [5] a proposé en 1996 un premier schéma *Bell laboratories LAYered Space Time* (BLAST) exploitant en même temps le multiplexage horizontal et vertical qui s'appelle le multiplexage spatio-temporel en blocs.

1.6.1 Multiplexage temporel

Cette technique consiste à transmettre chaque symbole d'information noté X_i où $i = \{0, 1, \dots, N-1\}$ indépendamment à divers moments T par une antenne émettrice. A titre d'exemple, nous allons considérer le cas où $N_t = N_r = 2$ et $T = 4$ dans la figure 1.16.

Dans cet exemple, on divise la trame en des sous-trames dépendants du nombre N_t . Puis, chaque sous-trame est envoyée à quatre instants différents $T = 4$. Ce type de multiplexage est connu sous le nom H-BLAST (Horizontal-BLAST). Il permet d'exploiter la diversité temporelle à l'émission.

1.6.2 Multiplexage spatial

Le multiplexage spatial permet de maximiser le débit d'information sur un canal MIMO. Cette technique a été introduite sous le nom de V-BLAST (Vertical-BLAST) dans [52] comme illustré dans la figure 1.17. Chaque symbole d'information S_i est transmis à un instant par une seule antenne de transmission. Comme dans le H-BLAST, on divise la trame en des sous-trames dépendants du nombre N_t . Mais dans ce cas, chaque sous-trame est envoyée à un seul instant $T = 1$.

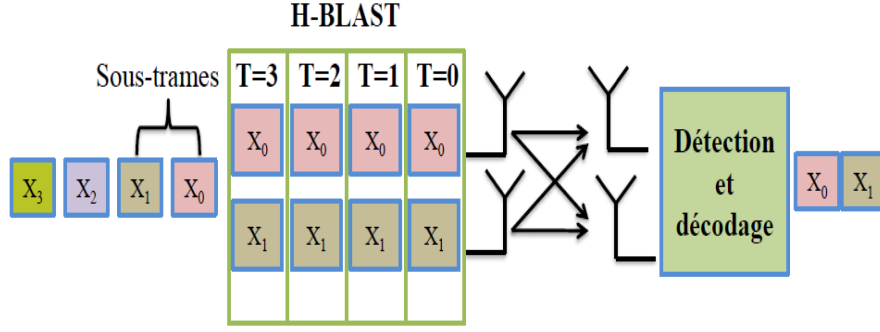


FIGURE 1.16 – Multiplexage temporel de type H-BLAST

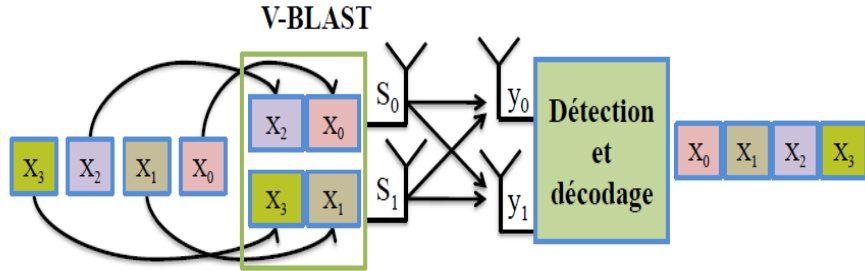


FIGURE 1.17 – Multiplexage spatial de type V-BLAST

Nous allons considérer le cas où $N_t = N_r = 2$ et $T = 1$ alors le rendement de ce code MIMO est $R_{MIMO} = 2$. La matrice de code est la suivante :

$$C_{V-BLAST} = \begin{bmatrix} S_0 \\ S_1 \end{bmatrix} \quad (1.30)$$

Le signal reçu s'écrit alors :

$$\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = H_c \begin{bmatrix} S_0 \\ S_1 \end{bmatrix} + \begin{bmatrix} n_0 \\ n_1 \end{bmatrix} \quad (1.31)$$

1.6.3 Codage spatio-temporel par blocs

Les codes spatio-temporels apportent une dépendance entre le domaine temporel et spatial afin d'augmenter l'efficacité spectrale et d'augmenter l'ordre de diversité. Son rôle consiste à répartir les bits ou les symboles modulés sur les différents degrés de liberté du canal. Les codes spatio-temporels se décomposent en deux familles : les codes spatio-temporels en treillis (STT) et les codes spatio-temporels en blocs (STB). Les codes spatio-temporels en blocs sont définis sous une forme matricielle. Ainsi, les symboles à transmettre sont codés par des opérations matricielles.

En 1998, Alamouti [53] a construit un code spatio-temporel orthogonal qui atteint la diversité maximale pour un système MIMO 2x1 avec un rendement égal à 1. L'avantage des codes orthogonaux est leur décodage linéaire. De plus, ils peuvent atteindre la diversité maximale. Par contre leur rendement reste limité à 1. Les codes STT sont une généralisation des modulations codées en treillis aux canaux MIMO [54]. Bien que les performances obtenues par ces codes soient excellentes,

la complexité du décodage est exponentielle selon le rendement. En effet, pour le décodage, on utilise une version vectorisée d'un décodage par treillis. Dans cette étude, nous ne présenterons que les codes spatio-temporels en blocs qui sont plus intéressants sur le plan pratique.

1.6.3.1 Codage de type Alamouti 2x1

Afin d'améliorer la qualité de la transmission, Alamouti [53] a conçu des systèmes basés essentiellement sur la diversité. Le code d'Alamouti est un code spatio-temporel orthogonal conçu pour un système MIMO avec deux antennes en émission $N_t = 2$ et une antenne en réception $N_r = 1$ (cas d'école). Ce code divise le mot de code en des couples. Puis, chaque couple est envoyé à deux instants différents $T = 2$ avec un rendement $R_{MIMO} = 1$. La matrice d'Alamouti 2×1 est la suivante :

$$C_{Alamouti2 \times 1} = \begin{bmatrix} S_0 & -S_1^* \\ S_1 & S_0^* \end{bmatrix} \quad (1.32)$$

Lors du premier instant, les symboles S_0 et S_1 sont transmis respectivement sur la première et la seconde antenne. Au second instant, le symbole conjugué du symbole S_1 noté S_1^* et l'inverse du symbole conjugué de S_0 noté S_0^* sont transmis. L'utilisation du code d'Alamouti $N_t = 2$ et $N_r = 1$ permet en l'associant à un code correcteur d'erreurs de s'approcher de la capacité d'un canal MIMO.

1.6.3.2 Codage de type Hadamard 2x2

Nous pouvons orienter les codes spatio-temporels vers d'autres matrices en appliquant la transformation de Hadamard. En 2002, Kaiser a utilisé les matrices de Hadamard qui minimisent les interférences entre symboles [55]. Par exemple, La construction d'une matrice 2x2 est donnée par :

$$C_{Hadamard2 \times 2} = \frac{1}{\sqrt{2}} \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} \quad (1.33)$$

Ces matrices sont très efficaces et leur utilisation conduit à une implantation simple de la tâche de précodage. Il s'agit d'une technique similaire à la transformée de Fourier qui permet de réaliser le précodage et son inverse en domaine logarithmique avec que des additions [56].

1.7 Techniques de détection à la réception de systèmes MIMO

Sur un canal de transmission MIMO, une stratégie efficace de décision consiste à choisir la séquence d'information la plus proche de la séquence reçue au sens de la distance euclidienne. Or, le récepteur sait calculer la valeur théorique des symboles qu'il peut recevoir.

Le rôle du détecteur est donc de comparer les séquences de symboles reçues aux séquences théoriques pré-calculées. Il existe différentes techniques de détection MIMO. Le choix d'une technique repose sur deux critères : les performances doivent être les plus proches possibles d'une détection optimale en termes de taux d'erreurs et la mise en oeuvre doit être la moins complexe possible.

1.7.1 Détection optimale MV (Maximum de Vraisemblance)

La détection à maximum de vraisemblance MV consiste à choisir la séquence d'information la plus proche de la séquence reçue au sens de la distance euclidienne [57].

$$\tilde{S}_i = \underset{S}{\operatorname{argmin}} ||Y - H_c.S||^2$$

Ce type de détection est optimal en termes de probabilité d'erreur, mais il présente une complexité rédhibitoire qui augmente exponentiellement avec la taille de la séquence reçue et l'ordre de modulation. Pour une séquence de longueur N et avec M états pour les symboles émis, le nombre de

séquences possibles d'information est de M^N .

Viterbi [28] a proposé un algorithme qui est capable de réduire le nombre d'opérations. En outre, il est possible d'appliquer en plus le décodage par sphère [58] afin de réduire encore l'espace de recherche. Ce décodage consiste à limiter l'espace de recherche des symboles possibles à une zone limitée. En revanche, la difficulté de cet algorithme est la détermination de l'espace de recherche concernant les symboles les plus fiables. Il existe un autre moyen de détection de plus faible complexité [6] qui sera introduit dans la suite.

1.7.2 Détection MV bit par bit des systèmes MIMO

Les données reçues par le récepteur sont traitées par les différents éléments de la chaîne de réception afin de restituer les données aussi proches que possible des informations émises. Dans un premier temps, les fonctions qui composent la détection travaillent sur des données binaires.

En outre, les informations nécessitent une représentation de la fiabilité de ces données. Chacun des éléments prend donc en entrée une information pondérée et restitue une sortie pondérée. On peut alors parler de traitement valeur LRV de chaque information binaire. Cela nécessite de rajouter des convertisseurs symbole/LRV et LRV/symbole.

1.7.2.1 Convertisseur Symbole vers LRV

Le convertisseur symbole/LRV [59] permet de convertir les symboles détectés \tilde{S}_i sous la forme de LRV. Ces symboles ont été préalablement fournis par le détecteur sous la forme :

$$\tilde{S}_i = h_{j,i}S_i + n_j$$

où $i = 0, 1, \dots, (N_t - 1)$ et $j = 0, 1, \dots, (N_r - 1)$. Avec $h_{j,i}$ est un coefficient canal et n_j est un bruit additif Gaussien. On obtient le LRV associé au bit d'indexe ii du symbole \tilde{S}_i en appliquant la formule suivante :

$$LRV_{[b_{ii}]} = \ln \left(\frac{\sum_{\substack{b_{ii}=0 \\ b_m \in S, (m \neq ii)}} P(Y|S, H_c) Pr(S \setminus b_{ii})}{\sum_{\substack{b_{ii}=1 \\ b_m \in S, (m \neq ii)}} P(Y|S, H_c) Pr(S \setminus b_{ii})} \right) \quad (1.34)$$

avec $ii = 0, 1, \dots, (\{N_t \log_2 M\} - 1)$ et $j = 0, 1, \dots, \{N_r - 1\}$

$$P(Y|S, H_c) = \frac{1}{\sqrt{\pi\sigma_w^2}} \exp \left(-\frac{1}{\sigma_w^2} \sum_{j=0}^{N_r-1} \left| y_j - \sum_{i=0}^{N_t-1} h_{j,i} S_i \right|^2 \right) \quad (1.35)$$

et

$$Pr(S \setminus b_{ii}) = Pr(b_m) \quad (1.36)$$

$Pr(S \setminus b_{ii})$ représente toutes les informations *a priori* hormis l'information *a priori* de b_{ii} .

1.7.2.2 Convertisseur LRV vers symbole

Le convertisseur LRV/symbole prend les LRV sur les bits détectés et fournit une estimation LRV du symbole correspondant [59]. L'estimation $LRV(\tilde{S})$ du symbole transmis S est donnée par :

$$LRV(\tilde{S}) = \sum_{b_{ii} \in S}^{\{\log_2 M\}} b_{ii} \times LRV(b_{ii}) \quad (1.37)$$

où $LRV(b_{ii})$ est l'ensemble des LRV a priori des bits détectés qui composent le symbole transmis S appartenant à la constellation χ de taille $\log_2 M$. En effet, le terme $Pr(S)$ désigne, la probabilité *a priori* d'avoir émis le symbole S . En supposant que les bits transmis sont statistiquement indépendants, cette probabilité devient :

$$Pr(S) = \prod_{ii=0}^{\{\log_2 M - 1\}} Pr(b_{ii}) \quad (1.38)$$

où b_{ii} sont les bits qui composent le symbole S selon la règle de *mapping*.

1.8 Performance du système MIMO 2x2 non itératif associé à un code LDPC non-binaire

Les performances du système de transmission MIMO (2x2) intégrant une détection basée sur le maximum de vraisemblance (MV) suivi d'un décodage de codes LDPC non-binaires sont présentées dans cette section selon différents paramètres. Dans notre étude nous considérons un canal Rayleigh non sélectif en fréquence et quasi-statique.

Rappelons qu'un canal quasi-statique reste constant durant la transmission d'une trame codée. Le canal SISO et le canal MIMO sont des variables aléatoires indépendantes et identiquement distribuées (i.i.d) qui suivent une loi de Rayleigh. Ce type de canal permet d'obtenir une modélisation réaliste d'une transmission MIMO et facilite sa mise en pratique. Les performances sont mesurées en terme de TES (Taux d'Erreur Symbole).

1.8.1 Performances SISO vs MIMO

La figure 1.18 présente le résultat de simulation pour un système SISO associé à un mapping 16-QAM et un système MIMO (2x2) associé à un mapping QPSK, ayant la même efficacité spectrale $\eta = 4\text{bps/Hz}$ pour une trame de longueur $N=2304$ bits et $R=1/2$. Sur cette figure, nous pouvons constater que pour la même efficacité spectrale du canal, la performance du canal SISO semble réellement moins bonne que celle du canal MIMO environ 2.5 db à 10^{-3} .

1.8.2 Performances système MIMO pour différents ordres de constellations

Sur la figure 1.19, les performances du récepteur pour un canal MIMO (2x2) quasi-statique aux cas des mappings QPSK, 16-QAM et 64-QAM suivent d'un multiplexage spatial sont présentées. On observe que la bien meilleure performance pour le mapping QPSK.

En effet, le niveau d'interférences générées par le multiplexage est d'autant plus difficile à traiter que l'ordre de mapping est élevé. En revanche, l'utilisation d'un mapping 64-QAM permet de multiplier l'efficacité spectrale par 6 et donc d'autant le débit binaire.

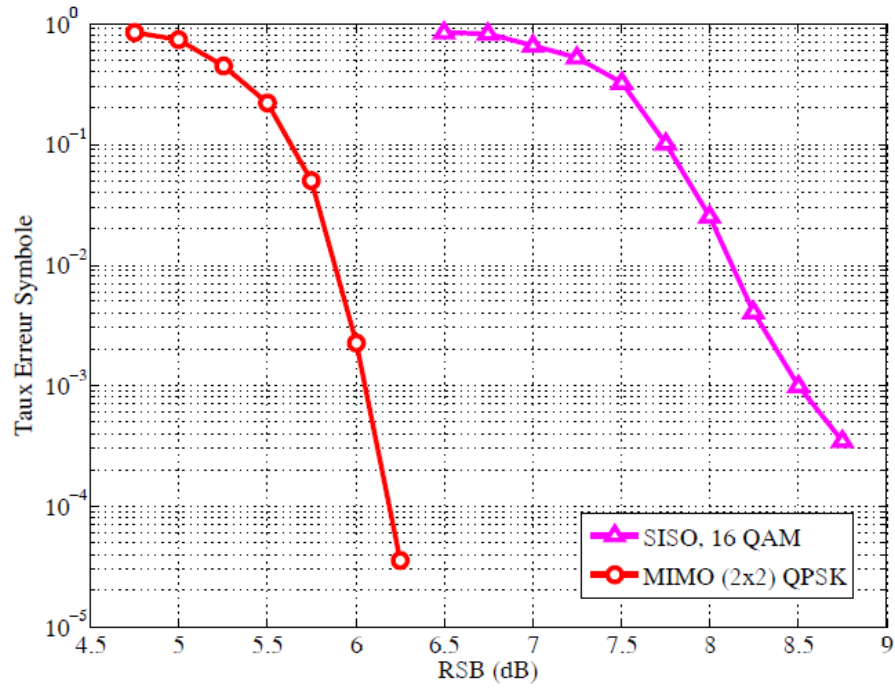


FIGURE 1.18 – Performances du récepteur pour un canal SISO et un canal MIMO 2x2 de même efficacité spectrale.

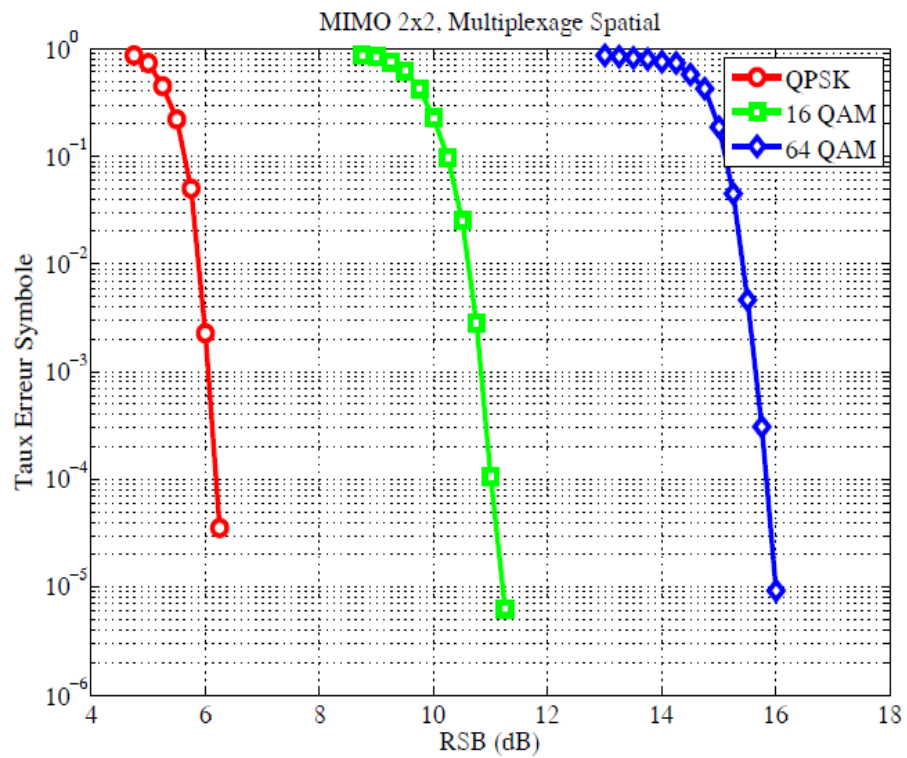


FIGURE 1.19 – Performances du récepteur pour un canal MIMO 2x2 avec les mappings QPSK, 16-QAM et 64-QAM

1.8.3 Performances système MIMO pour différents codes LDPC binaires et non-binaires

La figure 1.20 présente la comparaison en termes de performances d'erreur entre les codes LDPC non-binaires pour des mots de code de taille petite et modérée (288, 1152, 2304, 2880) $\in GF(64)$ et l'équivalent binaire $\in GF(2)$ dans le cas où $N = 2304$ bits qui correspond à la norme standard binaire Worldwide interoperability for Microwave Access (WiMAX) [60]. Nous remarquons qu'un gain important en termes de performances est obtenu en passant de $GF(2)$ à $GF(64)$.

On constate que le code LDPC non-binaire pour ($N = 2304$ bits) surpasse son équivalent binaire WIMAX d'environ ≈ 0.3 dB. En outre, un code LDPC non-binaire est plus performant que sa longueur de trame est plus grande. Cependant, l'augmentation de la longueur de trame implique une augmentation de complexité et de la latence au niveau du décodeur. C'est pourquoi, nous nous limitons la longueur de trame dans notre étude à ($N = 2304$ bits). Ces résultats sont les meilleurs connus à ce jour des codes à décodage itératif.

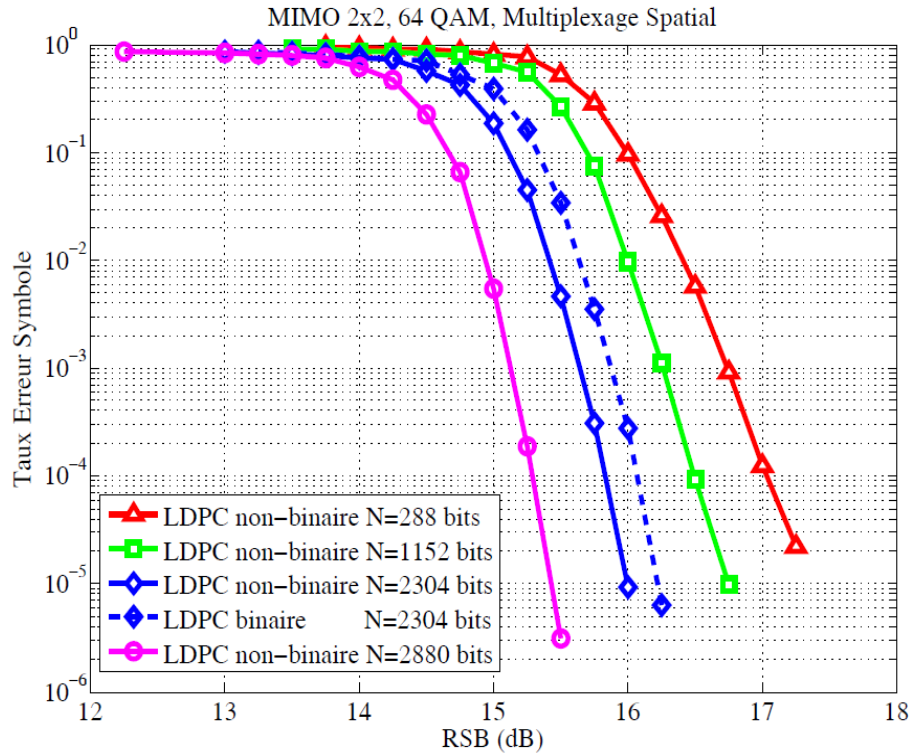


FIGURE 1.20 – Performances système MIMO pour un code LDPC non-binaire avec différents longueurs de trame et pour un code LDPC binaire WIMAX.

1.8.4 Performances système MIMO pour différentes techniques de transmissions MIMO

Les techniques de transmissions MIMO permettent d'exploiter la diversité d'une transmission sur un canal à évanouissements en créant des combinaisons linéaires entre les différents symboles d'information. La figure 1.21 illustre le gain apporté pour différentes techniques de transmissions MIMO (Hadamard, Alamouti et multiplexage spatial).

On note un gain de performance de 0.1 dB en utilisant le Hadamard par rapport au multi-

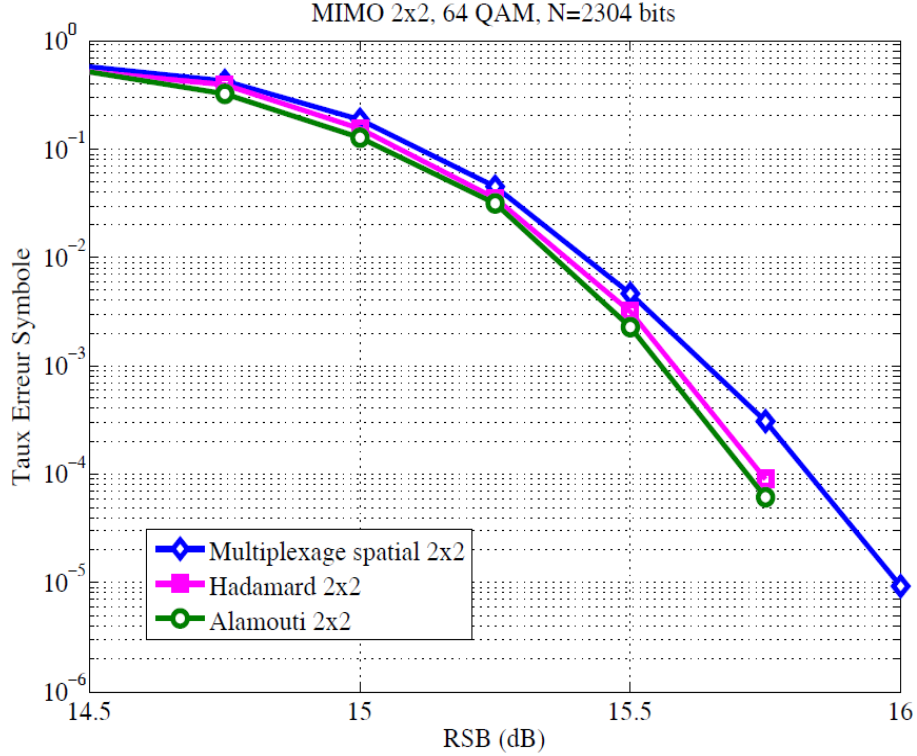


FIGURE 1.21 – Performances système MIMO pour différentes techniques de transmissions numériques.

plexage spatial. Encore, nous remarquons que les performances du système en utilisant le Hadamard et l'Alamouti sont quasiment identiques. Notons que l'Alamouti a un rendement $R = 1$ alors que le Hadamard et le multiplexage spatial ont un rendement $R = 2$ nommé (*full rate*).

Donc, l'Alamouti est moins efficace spectralement. En revanche, Hadamard (2x2) a une complexité très importante par rapport à un multiplexage spatial. En effet, le multiplexage spatial permet de maximiser le débit d'information sur un canal MIMO ayant une complexité raisonnable par rapport aux autres techniques de transmissions.

1.9 Conclusion

Ce chapitre a présenté les techniques de base introduisant de la diversité à l'émission et à la réception avec les systèmes MIMO. La liste n'est pas exhaustive et se limite aux techniques qui vont être étudiées dans ce mémoire. Une chaîne de transmission MIMO a été décrite dans ce premier chapitre. On a étudié les différents blocs de cette chaîne qui composée d'un codage de canal suivi par un mapping de divers ordres.

En plus, cette chaîne de transmission MIMO a été définie sous l'hypothèse que le canal est non sélectif en fréquence estimé par le récepteur. Ensuite, les principales techniques MIMO à l'émission et à la réception ont été aussi présentées. Dans ce contexte, le détecteur de l'algorithme de croyance et le détecteur MV avec information *a priori* ont été décrits.

Enfin, une comparaison de système MIMO pour différents paramètres a été réalisée afin de montrer les performances théoriques d'un tel récepteur MIMO sur un canal de Rayleigh. Le chapitre suivant sera consacré aux études algorithmiques de la détection MIMO.

Récepteur itératif MIMO basé sur la propagation de croyance

Dans ce chapitre nous décrivons un récepteur efficace pour les systèmes de communications numériques réalisant conjointement la détection d'un système MIMO et le décodage de codes LDPC au sein d'un processus itératif. Grâce à ce processus itératif, des informations extrinsèques sont échangées entre le détecteur et le décodeur afin d'approcher les performances optimales d'une transmission sans interférences.

Après une présentation du principe itératif selon un ordonnancement par inondation ("*flooding*"), nous proposons un schéma de réception MIMO admettant un détecteur sous-optimal basé sur l'algorithme de propagation de croyance étudié comme une solution alternative au détecteur à maximum de vraisemblance (MV). Puis, nous étudions des récepteurs itératifs reposant sur deux séquencements différents pour l'échange de messages entre le détecteur et le décodeur : "Séquencement *flooding*" et "Séquencement *shuffle*".

Enfin, nous procédons à une analyse théorique du récepteur proposé en termes de performances et de complexité calculatoire. La méthode des diagrammes EXIT (*EXtrinsic Information Transfer*) est appliquée afin d'analyser le comportement de convergence du détecteur MIMO basé sur la propagation de croyance proposée.

2.1 Principe itératif à l'égalisation

En 1995, C. Douillard *et al.* [4] ont proposé d'utiliser l'information issue du décodeur de canal pour améliorer la détection des symboles reçus. Par ce biais, le décodeur et le détecteur améliorent notablement leur estimation des données transmises au cours d'un processus itératif. Ce principe a été appliqué pour la première fois avec des turbo-codes, inventés par C. Berrou *et al.* [3]. Ce fût la première expérimentation efficace d'un principe de détection et de décodage conjoints en réception. Le traitement itératif utilisé dans les turbo-codes a été généralisé sous l'appellation de "principe turbo" ou "turbo-égalisation" [4]. Puis, une structure de turbo-égalisation adaptative a été proposée par A. Galvieux *et al.* en 1997 [61].

Dans notre étude, la turbo-égalisation est appliquée afin d'effectuer un échange d'informations extrinsèques entre un détecteur MIMO basé sur l'algorithme de propagation de croyance et un décodeur de canal à partir d'une information *a priori* pondérée dans le but d'améliorer la fiabilité de l'information au cours des itérations comme illustrée la figure 2.1. Dans le cadre de systèmes MIMO, le principe turbo a été utilisé pour traiter les interférences entre antennes. Ainsi, des gains impressionnants en termes de performance ont été obtenus par rapport à des systèmes classiques [62], [63], [64].

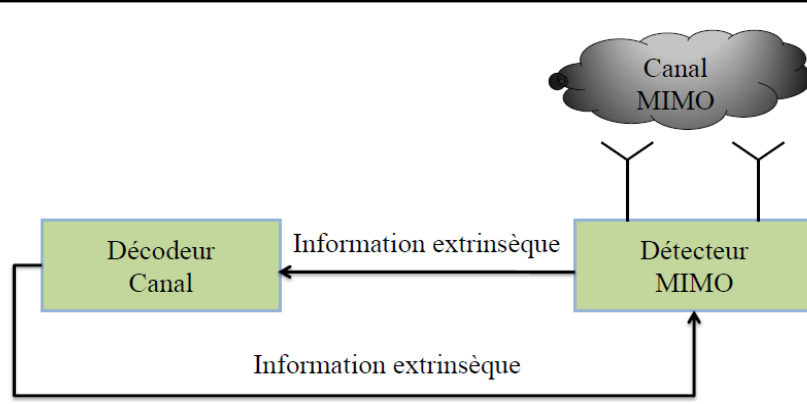


FIGURE 2.1 – Représentation du "principe turbo" ou "turbo-égalisation"

2.2 Système de réception itératif

Dans cette section, nous définissons les paramètres de notre système afin d'établir ses performances selon différents paramètres. Lors de cette étude, nous modélisons, comme illustré dans la figure 2.2, une chaîne de transmission MIMO intégrant un code LDPC-NB régulier défini sur $GF(2^6)$ associé à une constellation 64-QAM. Ce schéma a été choisi afin d'obtenir de meilleures performances pour des transmissions haut débit de données tels que présenté dans le premier chapitre. Nous avons choisi les paramètres du code LDPC-NB de sorte que le code LDPC binaire adopté par la norme IEEE WiMAX [60] soit utilisé comme référence. Donc, le code LDPC-NB a une longueur de trame $N = 384$ symboles, avec un rendement de codage $R = 1/2$. Les nœuds de variable ont un degré $d_v = 2$ et les nœuds de parité ont un degré $d_c = 4$ comme expliqué dans [45].

Un multiplexage spatial (SM) avec $N_t = 2$ antennes à l'émission et $N_r = 2$ antennes à la réception est appliqué. Chaque lien SISO du canal MIMO est une variable aléatoire indépendante et identiquement distribuée (i.i.d) qui suit une loi de Rayleigh. Des variables indépendantes et identiquement distribuées sont des variables aléatoires qui ont toutes la même loi de probabilité et sont mutuellement indépendantes. En outre, notre chaîne de réception MIMO intègre une détection basée sur l'algorithme BP suivie d'un décodage de codes LDPC-NB également basé sur cet algorithme. Les paramètres de notre système sont récapitulés dans le tableau 2.1. Dans la suite, nous étudions ses

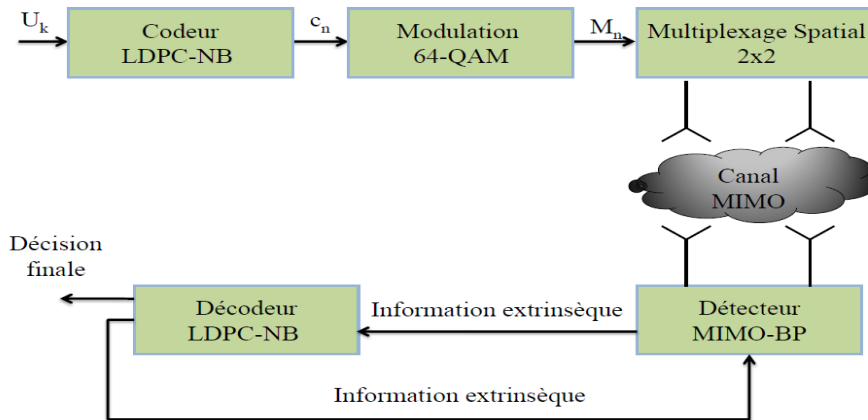


FIGURE 2.2 – Chaîne de communications numériques étudiée

Code correcteur d'erreurs	LDPC non-binaire
Taille de la trame	(384 symboles $\in GF(64)$) ou 2304 bits
Modulation	64 – QAM
Précodage linéaire	Multiplexage spatial 2x2
Type de canal	Rayleigh i.i.d
Canal	MIMO 2x2
Détection	algorithme BP
Décodage	algorithme BP

TABLE 2.1 – Paramètres du système itératif MIMO basé sur la propagation de croyance

performances selon différentes configurations.

2.3 Représentation conjointe des graphes associés au détecteur et au décodeur

La proposition d'un traitement itératif en réception basé sur les graphes bipartites du détecteur MIMO et du décodeur de code LDPC doit permettre la conception de récepteurs flexibles caractérisés par de faibles latences. De plus, la réalisation d'un détecteur et d'un décodeur d'une manière disjointe s'accompagne d'une perte d'information. Dans une approche basée sur la propagation de croyance, le détecteur MIMO et le décodeur LDPC-NB peuvent être combinés pour aboutir à un graphe conjoint détecteur-décodeur, dit GFC (Graphe à Facteur Conjoint). Ce dernier permet d'envisager différents séquencements. Notons que le terme "séquencement" correspond à un algorithme de passage de messages au niveau récepteur qui favorise l'échange d'informations entre le détecteur et le décodeur. Tandis que le terme "ordonnancement" correspond à un algorithme de passage de messages au sein du détecteur et au sein du décodeur.

Le graphe conjoint GFC du détecteur MIMO basé sur la propagation de croyance et du décodeur LDPC-NB est donné dans la figure 2.3. Ce graphe se compose de deux parties disjointes. La

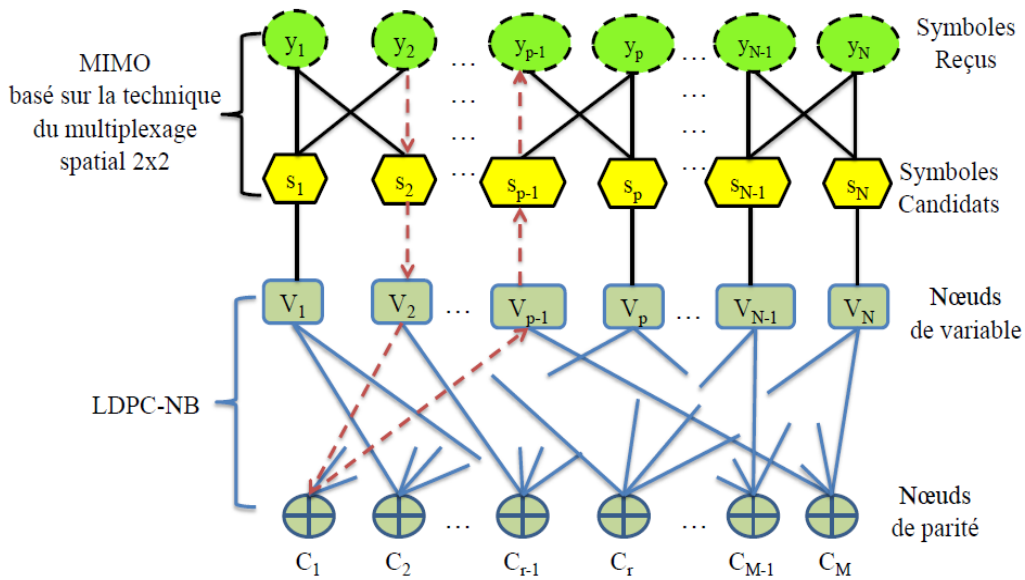


FIGURE 2.3 – Représentation d'un graphe conjoint GFC

partie supérieure du graphe est la représentation graphique d'un détecteur MIMO basé sur la technique du multiplexage spatial composé de $N_t = N_r = 2$, en illustrant les symboles reçus et les symboles candidats. La partie inférieure du graphe correspond à la représentation graphique de la matrice de décodage d'un code LDPC-NB en indiquant les nœuds de variable et les nœuds de parité.

Dans des travaux récents [65], [66], un graphe conjoint de type GFC a été proposé pour la détection MIMO et le décodage de canal. Dans cette étude préalable, seul le séquençement de type *flooding* a été exploré où chaque couche du GFC est traitée successivement. En appliquant le séquençement de type *flooding*, le détecteur fonctionne sur toute la trame reçue, puis génère les informations souples pour le décodeur. Par contre, lors d'utilisation du séquençement de type *shuffle*, l'information extrinsèque est échangée entre le détecteur et le décodeur au cours d'une inter-itération. Par conséquent, son information extrinsèque peut être directement renvoyée comme information *a priori* pour une nouvelle détection d'un symbole.

Dans notre contexte, les deux séquençements *flooding* et *shuffle* sont appliqués au graphe GFC au sein du récepteur itératif. Plusieurs ordonnancements efficaces au sein du décodeur LDPC ont été proposées dans la littérature. En effet, la matrice de parité peut décoder séquentiellement les messages par un ordonnancement *flooding* ou *shuffle*. Le décodage itératif peut ensuite être divisé en sous-couche au sein d'une itération. L'ordonnancement *shuffle* permet d'éviter la latence au cours d'un décodage itératif [67]. L'idée de base est d'exécuter toutes les sous-couches du décodeur en parallèle et d'échanger des informations extrinsèques dès qu'elles sont disponibles. Dans notre étude, nous avons appliqué un ordonnancement *shuffle* au sein du décodeur LDPC-NB afin d'éviter la latence au sein de notre système itératif et d'échanger les informations extrinsèques au sein du décodeur.

Dans notre étude, chaque couple de nœuds de symboles reçus est relié au couple de nœuds symboles candidats qui ont le même indice, grâce à un multiplexage spatial avec $N_t = 2$ antennes d'émission et $N_r = 2$ antennes de réception. D'un autre côté, le $p^{\text{ème}}$ symbole candidat, ($p = 1, 2, \dots, N$), est connecté au nœud de variable du graphe bipartite du code LDPC-NB qui a le même indice. Les connexions entre les nœuds de variable et les nœuds de parité, ($r = 1, 2, \dots, M$), dépendent des contraintes de la matrice de parité du code LDPC-NB contenant ($M = N - K$) nœuds de parité.

Dans un premier temps, nous allons présenter l'algorithme de propagation de croyance en détectant les symboles reçus bit par bit. Dans un second temps, nous présenterons la détection non-binaire des symboles reçus suivant l'algorithme BP.

2.3.1 Détection bit par bit lors de la propagation de croyance

L'équation (1.23)

$$y_j = \sum_{i=1}^{N_t} h_{ji} x_i + n_j \quad (2.1)$$

donne l'expression du vecteur de données reçu dans le cas général d'un système MIMO pour une distribution de Rayleigh. Notre but est de détecter les symboles reçus en fournissant une estimation la plus fiable possible des symboles transmis. Le signal reçu y_j à chaque instant t sur la $j^{\text{ème}}$ antenne de réception est la somme des symboles bruités issus des N_t signaux transmis.

Dans cette étude, la chaîne de réception MIMO intègre une détection basée sur l'algorithme BP suivie d'un décodage de codes LDPC-NB également basé sur l'algorithme BP. Il s'agit d'un algorithme itératif faisant partie de la classe des algorithmes sous-optimaux. Il calcule le vecteur des

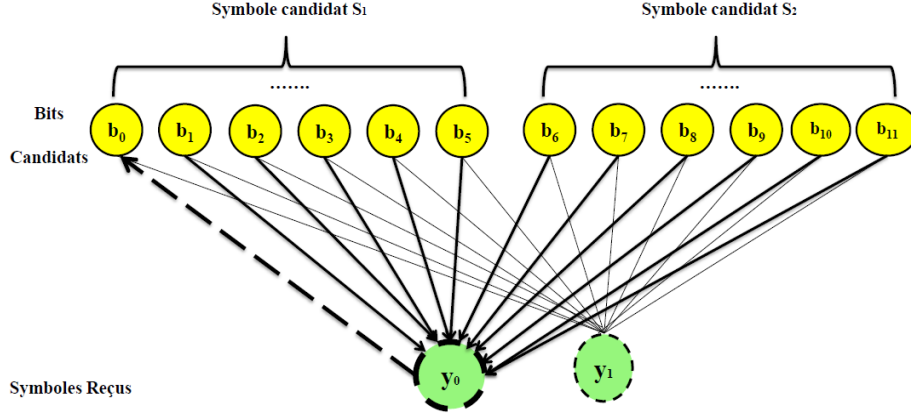


FIGURE 2.4 – Etape 1 de la détection MIMO basé sur la propagation de croyance binaire

informations souples des LRV des symboles candidats S_i devant être envoyés au décodeur LDPC-NB. Ce détecteur dispose d'une information *a priori* sur les données transmises sous forme de LRV en supposant la connaissance de symbole reçu y_j qui est composé de $(q = \log_2 M)$ bits et celles des coefficients du canal H_c .

Tout d'abord, tous les vecteurs des messages LRV *a priori* associés aux bits b_i sont initialisés à zéro. Ensuite, les étapes effectuées lors de l'exécution d'une itération sont les suivantes :

1-) La première étape, illustrée dans la figure 2.4, consiste à envoyer les messages LRV *a priori* des $i^{\text{èmes}}$ bits candidats b_i au $j^{\text{ème}}$ symbole reçu y_j . Dans le nœud y_j , le message LRV de chaque bit candidat b_i est mis à jour par le calcul de toutes les distances euclidiennes entre la séquence reçue à la $j^{\text{ème}}$ antenne et les séquences candidates, en tenant compte des informations *a priori* des autres bits candidats b_i . Les messages LRV, recalculés par le nœud du symbole reçu y_j , sont envoyés au $i^{\text{ème}}$ bit candidat b_i tels que :

$$L_{y_j \rightarrow b_i}^{(t)} = \log \left(\frac{\sum_{b_m \in S, b_i=0} P(y_j|S, H_c) Pr(S \setminus b_i)}{\sum_{b_m \in S, b_i=1} P(y_j|S, H_c) Pr(S \setminus b_i)} \right) \quad (2.2)$$

où $\{j = 1, \dots, N_r\}$ et $\{i = 1, 2, \dots, (N_t \times \log_2 M)\}$. b_m représente tous les bits candidats connectés au symbole reçu y_j à l'exclusion de b_i , avec

$$P(y_j|S, H_c) = \frac{1}{\sqrt{\pi\sigma_w^2}} \exp \left(-\frac{1}{\sigma_w^2} \left| y_j - \sum_i h_{j,i} \cdot S_i \right|^2 \right) \quad (2.3)$$

$Pr(S \setminus b_i)$ représente toutes les informations *a priori* à l'exception des informations *a priori* de b_i . Pour simplifier le calcul de l'équation (2.2), l'approximation max-log ($\log(e^x + e^y) \approx \max(x, y)$) est appliquée. L'équation (2.2) est ainsi simplifiée comme suit :

$$L_{y_j \rightarrow b_i}^{(t)} \approx \max_{S: b_i=0} \left\{ -\frac{1}{\sigma_w^2} \left| y_j - \sum_i h_{j,i} \cdot S_i \right|^2 + \sum_{k \neq j} L_{y_k \rightarrow b_m}^{(t-1)} + \sum_{b_m \in S, m \neq i} L_{P \rightarrow b_m}^{(t-1)} \right\} \\ - \max_{S: b_i=1} \left\{ -\frac{1}{\sigma_w^2} \left| y_j - \sum_i h_{j,i} \cdot S_i \right|^2 + \sum_{k \neq j} L_{y_k \rightarrow b_m}^{(t-1)} + \sum_{b_m \in S, m \neq i} L_{P \rightarrow b_m}^{(t-1)} \right\} \quad (2.4)$$

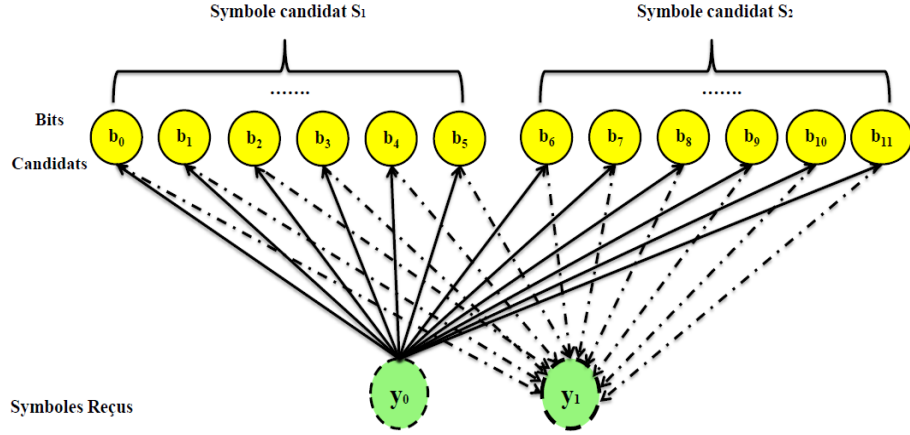


FIGURE 2.5 – Etape 2 de la détection MIMO basé sur la propagation de croyance binaire

où $L_{P \rightarrow b_m}^{(t-1)}$ représente l'information extrinsèque provenant du décodeur LDPC durant l'itération précédente. $\sum_{k \neq j} L_{y_k \rightarrow b_m}^{(t-1)}$ correspond aux informations extrinsèques provenant des autres bits candidats.

2-) Durant la seconde étape, illustrée dans la figure 2.5, les bits candidats b_i sont mis à jour à partir des informations provenant du symbole y_j et des informations *a priori* provenant du décodeur lors de l'itération précédente. Ainsi, les bits candidats b_i sont mis à jour de la manière suivante :

$$L_{b_i}^{(t)} = \sum_{j=1}^{N_r} L_{y_j \rightarrow b_i}^{(t)} + L_{P \rightarrow b_i}^{(t-1)} \quad (2.5)$$

où $L_{y_j \rightarrow b_i}^{(t)}$ désigne les messages LRV échangés entre le $j^{\text{ème}}$ symbole reçu y_j et le $i^{\text{ème}}$ bit candidat b_i . Les nouveaux messages LRV des bits candidats sont utilisés comme des nouvelles informations *a priori* pour le nouveau calcul des vecteurs LRV associés au $(j+1)^{\text{ème}}$ symbole reçu. Les messages LRV calculés par le bit candidat b_i sont envoyés au nœud du symbole reçu y_j :

$$L_{b_i \rightarrow y_j}^{(t)} = \sum_{k=1|k \neq j}^{N_r} L_{y_k \rightarrow b_i}^{(t)} + L_{P \rightarrow b_i}^{(t-1)} \quad (2.6)$$

$$= L_{b_i}^{(t)} - L_{y_j \rightarrow b_i}^{(t)} \quad (2.7)$$

Finalement, les bits candidats sont mis à jour par les informations provenant du symbole y_{j+1} et les informations *a priori*. Plusieurs itérations au sein du détecteur MIMO basé sur la propagation de croyance sont nécessaires pour échanger les informations extrinsèques au sein du détecteur afin d'approcher les performances optimales d'une transmission sans interférences.

À l'issue de la détection, les informations souples associées aux bits sont envoyées au décodeur LDPC non-binaire. Le processus de décodage est appliqué selon un ordonnancement *shuffle*, processus de décodage par nœud de variable, jusqu'à convergence vers un mot de code ou atteinte d'un nombre maximal (prédéfini) d'itérations. Après le décodage, une information extrinsèque est renvoyée vers le détecteur. Les messages provenant du décodeur sont exploités comme des informations *a priori* pour une nouvelle phase de détection.

2.3.2 Détection non-binaire par propagation de croyance

Dans cette section, nous allons présenter une détection MIMO non-binaire basée sur l'algorithme de propagation de croyance. L'algorithme BP a été appliqué avec succès pour une détection non-binaire par symbole associée à un décodage canal dans [68]. Il a été montré dans [69] que ce type d'algorithme permet d'avoir une détection avec une complexité linéaire en fonction du nombre d'antennes contrairement à l'algorithme MV qui croît exponentiellement avec le nombre d'antennes.

Par conséquent, si on peut aboutir à la réduction de la complexité d'une détection non-binaire itérative, il est possible d'atteindre un gain significatif en terme de performance en optimisant les échanges de messages avec une complexité raisonnable. L'algorithme BP calcule l'information souple pour tous les symboles. Notons que le vecteur LRV d'un symbole non-binaire correspond à un mot de code c_i dans $GF(64)$. Il est défini comme suit :

$$L_{c_i} = [L_{[c_i, \alpha_0]}; L_{[c_i, \alpha_1]}; \dots L_{[c_i, \alpha_{63}]}]^T \quad (2.8)$$

avec chaque élément ayant pour expression :

$$L_{[c_i, \alpha_k]} = \ln \left(\frac{P(c_i = \alpha_k | y)}{P(c_i = \alpha_0 | y)} \right) \quad (2.9)$$

où $P(c_i = \alpha_k | y)$ représente la probabilité que le mot de code c_i soit égal à $\alpha_k \in GF(64)$. Ainsi, $L_{[c_i, \alpha_0]} = 0$ et $L_{[c_i, \alpha_k]} \in \mathbb{R}$. Chaque vecteur LRV correspond à un unique symbole non-binaire.

Dans notre étude, l'algorithme BP permet de calculer un vecteur d'informations souples LRV des symboles candidats notés S_i ($i = 1, 2, \dots, N$). Ce dernier est envoyé au décodeur LDPC-NB. Nous traitons les symboles par couple au sein du récepteur grâce à un multiplexage spatial avec $N_t = 2$ antennes d'émission et $N_r = 2$ antennes de réception. Le graphe de détection d'un couple de nœuds de symboles reçus associé à un couple de nœuds symboles candidats est donnée dans la figure 2.6.

En premier lieu, tous les messages *a priori* des vecteurs LRV associés aux symboles S_i sont initialisés à zéro. Puis, trois étapes sont successivement appliquées lors de l'exécution d'une itération comme le montre la Figure 2.6.

1-) L'étape 1 consiste à envoyer les vecteurs LRV *a priori* des $i^{\text{èmes}}$ symboles candidats S_i au $j^{\text{ème}}$ symbole reçu y_j . Dans le nœud y_j , le vecteur LRV pour chaque symbole candidat S_i est mis à jour par le calcul de toutes les distances euclidiennes entre la séquence reçue à la $j^{\text{ème}}$ antenne et les séquences candidates. Ce calcul tient compte des informations *a priori* des autres symboles candidats S_i . Puis, les vecteurs LRV sont envoyés du $j^{\text{ème}}$ symbole reçu y_j aux $i^{\text{èmes}}$ symboles candidats S_i . Ces vecteurs ont pour expression :

$$L^{(t)}_{y_j \rightarrow S_i} = [L_{[S_i, \alpha_0]}; L_{[S_i, \alpha_1]}; \dots L_{[S_i, \alpha_{63}]}]^T \quad (2.10)$$

où $L_{[S_i, \alpha_k]}$, $k = 0, 1, 2, \dots, 63$, représente la fiabilité pour le symbole candidat S_i d'être égal à $\alpha_k \in GF(64)$ tel que :

$$L_{[S_i, \alpha_k]} = \log \left(\frac{\sum_{\substack{S_m \in GF(64), (m \neq i) \\ S_m = \alpha_k}} P(y_j | S, \mathbf{H}) Pr(S \setminus S_i)}{\sum_{\substack{S_m \in GF(64), (m \neq i) \\ S_m = \alpha_0}} P(y_j | S, \mathbf{H}) Pr(S \setminus S_i)} \right) \quad (2.11)$$

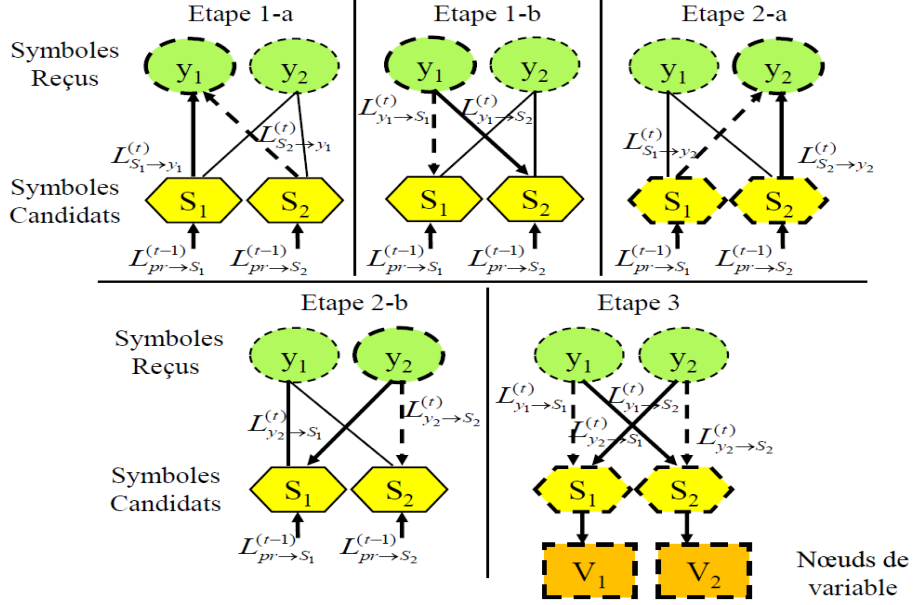


FIGURE 2.6 – Récepteur itératif MIMO basé sur la propagation de croyance reposant sur un traitement symbole

où $j = 1, 2, \dots, N_r$, $i = 1, 2, \dots, N_t$, $m = 1, 2, \dots, N_t$. S_m représente l'ensemble des symboles candidats connectés au symbole reçu y_j à l'exclusion de S_i , tel que :

$$P(y_j|S, \mathbf{H}) = \frac{1}{\sqrt{\pi\sigma_w^2}} \exp \left(-\frac{1}{\sigma_w^2} \left| y_j - \sum_i h_{j,i} \cdot S_i \right|^2 \right) \quad (2.12)$$

et

$$Pr(S \setminus S_i) = \prod_{\substack{(m=1|m \neq i) \\ S_m \in GF(64)}}^{N_t} Pr(S_m) \quad (2.13)$$

La probabilité $Pr(S \setminus S_i)$ représente toutes les informations *a priori* à l'exception des informations *a priori* de S_i . Elle peut être réécrite comme suit :

$$Pr(S \setminus S_i) = \prod_{\substack{m=1 \\ (m \neq i)}}^{N_t} Pr(S_m = \alpha_0) \exp \left(\sum_{\substack{(m=1|m \neq i) \\ \alpha_k \in GF(64)}}^{N_t} L_{[S_m, \alpha_k]} \right) \quad (2.14)$$

En utilisant (2.12) et (2.14), une nouvelle expression de l'équation (2.11) peut être obtenue :

$$L_{[S_i, \alpha_k]} = \log \left(\frac{\sum_{\substack{S_i = \alpha_k \\ S_m \in GF(64), (m \neq i)}} \exp(\mu)}{\sum_{\substack{S_i = \alpha_0 \\ S_m \in GF(64), (m \neq i)}} \exp(\mu)} \right) \quad (2.15)$$

$$\text{où } \mu = -\frac{1}{\sigma_w^2} \left| y_j - \sum_i h_{j,i} \cdot S_i \right|^2 + \sum_{\substack{(m=1|m \neq i) \\ \alpha_k \in GF(64)}}^{N_t} L_{[S_m, \alpha_k]}.$$

Pour simplifier le calcul de l'équation (2.15), l'approximation max-log $\log(e^x + e^y) \approx \max(x, y)$ est

appliquée. L'équation (2.15) est alors simplifiée comme suit :

$$L_{[S_i, \alpha_k]} \approx \max_{\substack{S_i = \alpha_k \\ S_m \in GF(64), (m \neq i)}} \left\{ \mu + \sum_{\substack{(m=1|m \neq i) \\ \alpha_k \in GF(64)}}^{N_t} L_{[S_m, \alpha_k]} \right\} - \max_{\substack{S_i = \alpha_0 \\ S_m \in GF(64), (m \neq i)}} \left\{ \mu + \sum_{\substack{(m=1|m \neq i) \\ \alpha_k \in GF(64)}}^{N_t} L_{[S_m, \alpha_k]} \right\} \quad (2.16)$$

2-) A l'issue de l'étape 1, les symboles candidats S_i sont mis à jour.

$$L_{S_i}^{(t)} = \sum_{k=1}^{N_r} L_{y_k \rightarrow S_i}^{(t)} + L_{pr \rightarrow S_i}^{(t-1)} \quad (2.17)$$

où $\sum_{k=1}^{N_r} L_{y_k \rightarrow S_i}^{(t)}$ désigne les vecteurs LRV échangés entre le $j^{\text{ème}}$ symbole reçu y_j et le $i^{\text{ème}}$ symbole candidat S_i . $L_{pr \rightarrow S_i}^{(t-1)}$ représente le vecteur LRV *a priori* de l'information extrinsèque provenant du décodeur LDPC-NB au cours de l'itération précédente. Durant l'étape 2, le message échangé entre le symbole S_i et le symbole reçu connecté y_j peut être exprimé par :

$$L_{S_i \rightarrow y_j}^{(t)} = \sum_{\substack{k=1 \\ k \neq j}}^{N_r} L_{y_k \rightarrow S_i}^{(t)} + L_{pr \rightarrow S_i}^{(t-1)} = L_{S_i}^{(t)} - L_{y_j \rightarrow S_i}^{(t)} \quad (2.18)$$

où $\sum_{\substack{k=1 \\ k \neq j}}^{N_r} L_{y_k \rightarrow S_i}^{(t)}$ désigne les vecteurs d'informations extrinsèques LRV des autres symboles reçus à l'exclusion du symbol y_j . $L_{S_i}^{(t)}$ représente l'information *a posteriori* du symbol S_i . Le vecteur LRV $L_{S_i \rightarrow y_j}^{(t)}$ est utilisé comme une nouvelle information *a priori* pour le nouveau calcul des vecteurs LRV associés au $(j+1)^{\text{ème}}$ symbole reçu. Puis, le calcul de l'équation (2.15) est appliqué afin de calculer le $L_{y_j \rightarrow S_i}^{(t)}$ associé au $(j+1)^{\text{ème}}$ symbole reçu.

Plusieurs intra-itérations au sein du détecteur MIMO basé sur la propagation de croyance sont effectuées par l'exécution des étapes 1 et 2 avant de mettre à jour les symboles candidats au cours d'un étape supplémentaire (étape 3).

3-) Finalement, les symboles candidats sont mis à jour à partir des informations provenant des $j^{\text{èmes}}$ symboles reçus et des informations extrinsèques provenant du décodeur LDPC-NB au cours de l'itération précédente :

$$L_{S_i}^{(t)} = \sum_{k=1}^{N_r} L_{y_k \rightarrow S_i}^{(t)} + L_{pr \rightarrow S_i}^{(t-1)} \quad (2.19)$$

Les informations souples associées aux symboles sont ensuite envoyées au décodeur LDPC-NB :

$$L_{pr \rightarrow V_p}^{(t)} = L_{S_i}^{(t)} - L_{pr \rightarrow S_i}^{(t-1)} \quad (2.20)$$

Le processus de décodage est itéré selon un ordonnancement *shuffle* [49] jusqu'à convergence vers un mot de code ou atteinte d'un nombre maximal (prédéfini) d'itérations. Après le décodage, une information extrinsèque est renvoyée vers le détecteur. Les messages provenant du décodeur sont exploités comme des informations *a priori* pour un nouveau traitement.

2.3.3 Décodage non-binaire basé sur la propagation de croyance

Comme mentionné dans le premier chapitre, les codes LDPC introduits par Gallager en 1962 ont ensuite été étudiés à la fin des années 1990. En 1998, Davey et Mackay ont proposés une version non-binaire des codes LDPC. Ces codes sont capables de surpasser les codes LDPC binaires pour des trames de longueurs courtes et moyennes [9].

Ce type de codes peut être facilement combiné avec des modulations d'ordres élevés et des systèmes à antennes multiples. Ainsi, des transmissions à débit de données très élevé [40] peuvent être supportées. Le décodage de codes LDPC-NB est un décodage itératif basé sur le principe de passage de messages. Deux principales étapes sont distinguées.

1-) Dans un premier temps, les messages correspondant aux nœuds de variable sont transmis à l'ensemble des nœuds de parité pour une mise à jour du calcul des parités.

2-) Dans un second temps, les messages de nœuds de parité sont transmis vers les nœuds de variable pour des calculs au sein des nœuds de variables.

Le principe de cet algorithme de décodage est que le message sortant sur la $(j)^{\text{ème}}$ branche est une fonction de tous les messages entrants sur ce nœud sauf celui de la $(j)^{\text{ème}}$ branche. Ainsi, les messages envoyés par un nœud de variable V à un nœud de parité C ne prends pas en compte le message échangé de V vers C lors de l'itération. Comme nous avons mentionné au début de ce chapitre, nous appliquons au sein du décodeur un ordonnancement *shuffle* afin d'éviter la latence au sein de notre système itératif et d'échanger les informations extrinsèques dès qu'elles sont disponibles.

Généralement, le processus itératif de décodage pour les codes LDPC-NB est composé de trois étapes. Au cours d'une étape préliminaire, les messages entrants provenant de tous les nœuds de parité connectés sont initialisés à zéro :

1-) La première étape consiste à mettre à jour les nœuds de variable V_p , $p \in 1, 2, \dots, N$, profitant de l'information extrinsèque provenant du détecteur MIMO basé sur la propagation de croyance et des informations issues des nœuds de parité C_r , $r \in 1, 2, \dots, M$ connectés à V_p au sein du décodeur LDPC-NB. Ainsi, le vecteur LRV du $p^{\text{ème}}$ nœud de variable à la $t^{\text{ème}}$ iteration, a pour expression :

$$L_{V_p}^{(t)} = \sum_{l=1}^{d_v} L_{l_{(p,r)} \rightarrow V_p}^{(t)} + L_{pr \rightarrow V_p}^{(t)} \quad (2.21)$$

où $L_{l_{(p,l)} \rightarrow V_p}^{(t)}$ correspond au message issu de la branche $l_{(p,r)}$ reliant le nœud de variable V_p au nœud de parité C_r . $L_{pr \rightarrow V_p}^{(t)}$ est l'information *a priori* provenant du détecteur MIMO basé sur la propagation de croyance.

2-) Durant la deuxième étape, l'information *a posteriori* du nœud de variable V_p est transmise aux $d_v - 1$ nœud de parité qui lui sont associés :

$$L_{V_p \rightarrow l_{(p,r)}}^{(t)} = \sum_{\substack{l=1 \\ l \neq r}}^{d_v} L_{l_{(p,r)} \rightarrow V_p}^{(t)} + L_{pr \rightarrow V_p}^{(t)} \quad (2.22)$$

3-) La mise à jour des nœuds de parité C_r connectés au nœud de variable V_p est effectuée au cours de la troisième étape. Puis, les informations extrinsèques sont renvoyés aux $d_c - 1$ branches

connectés à C_r . Ce processus est répété pour chaque nœud de variable.

Dans cette étude, nous avons choisi l'algorithme de décodage Min-Somme (MS) étendu aussi appelé (EMS) [44]. Ce dernier offre un bon compromis entre les performances et la complexité calculatoire pour traiter le graphe bipartite au sein des nœuds de parité. Dans la suite, nous allons détailler les séquencements *flooding* et *shuffle* pouvant être appliqués au graphe conjoint GFC du récepteur MIMO proposé basé sur la propagation de croyance.

2.4 Différents séquencements pour le système de réception

2.4.1 Séquencement *flooding*

Lorsqu'un séquencement *flooding* est appliqué, chaque couche du GFC est traitée successivement. En d'autres termes, le détecteur MIMO 2x2 basé sur la propagation de croyance fonctionne sur toute la trame reçue, comme illustré dans la figure 2.7, puis génère les informations souples pour le décodeur LDPC-NB.

Pour ce faire, nous appliquons la méthode décrite dans la section 2.3.2 à chaque couple de symboles reçus et sur toute la trame. Dans notre cas, comme l'algorithme BP est utilisé au sein du détecteur MIMO, plusieurs itérations internes au sein du détecteur appelée "intra-itérations" sont nécessaires avant de transmettre les informations souples au décodeur.

À la fin de ces intra-itérations au sein du détecteur, l'ensemble des informations souples associées à tous les symboles de la trame sont envoyées simultanément au décodeur LDPC-NB. Le processus de décodage est itéré selon un ordonnancement *shuffle* jusqu'à la convergence vers un mot

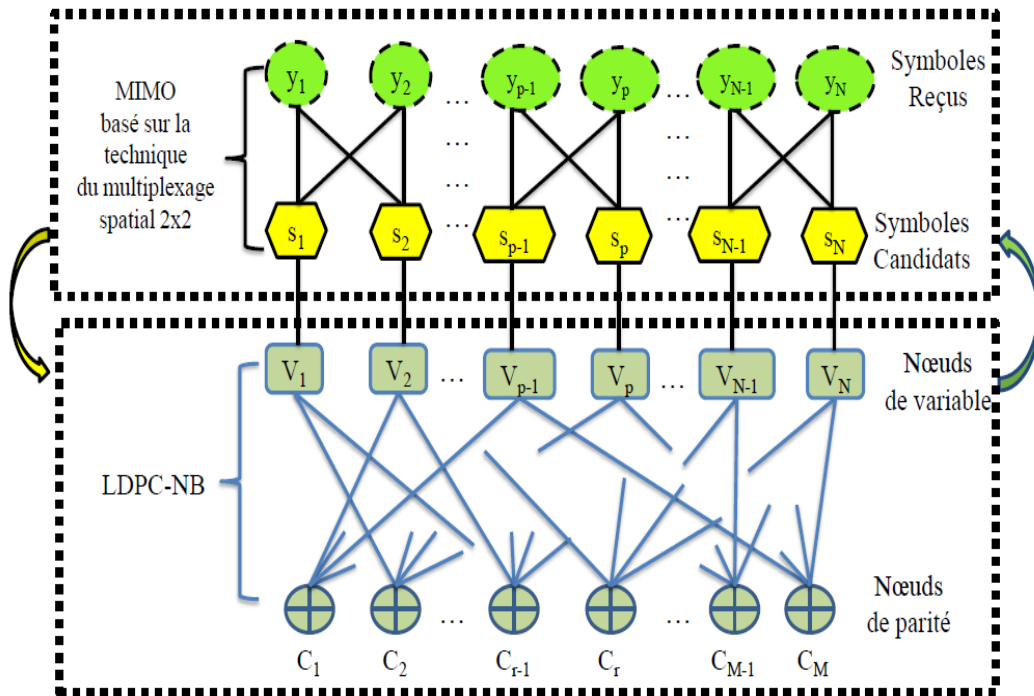


FIGURE 2.7 – séquencement *flooding* du GFC

de code ou la fin du nombre des itérations internes (intra-itérations). Le décodage est appliqué à toutes les nœuds de variable, en exploitant de l'information *a priori* provenant du détecteur MIMO basé sur la propagation de croyance.

Puis, les informations extrinsèques issues du décodeur LDPC-NB sont renvoyées vers le détecteur MIMO basé sur la propagation de croyance. Les messages provenant du décodeur sont exploités comme des informations *a priori* pour une nouvelle phase de détection. Ce processus correspond à un traitement itératif de type turbo-détection entre deux blocs indépendants. L'ensemble de ces deux étapes constitue une itération globale ici appelée "inter-itération".

2.4.2 Séquencement *shuffle*

Il est également possible d'appliquer un séquencement de type *shuffle* au sein du récepteur. Lors de l'utilisation de ce séquencement *shuffle*, l'information extrinsèque est échangée entre le détecteur MIMO basé sur la propagation de croyance et le décodeur LDPC-NB au cours d'une inter-itération. En effet, chaque nœud de variable peut être mis à jour de manière indépendante.

Par conséquent son information extrinsèque peut être directement renvoyée comme information *a priori* pour une nouvelle détection d'un symbole. Dans le système MIMO avec deux antennes à la réception, nous proposons de réaliser la détection MIMO sur les couples d'observations reçues. L'information souple générée est ainsi envoyée à un couple de nœuds de variable du décodeur LDPC-NB. Au cours de l'étape de décodage, l'algorithme BP est appliqué sur ces deux nœuds de variable qui sont mis à jour par les nœuds de parité connectés. Puis, l'information extrinsèque est renvoyée aux nœuds correspondants de la partie détection au sein du graphe GFC.

L'intérêt majeur d'un séquencement *shuffle* sur le graphe GFC d'un récepteur itératif est de

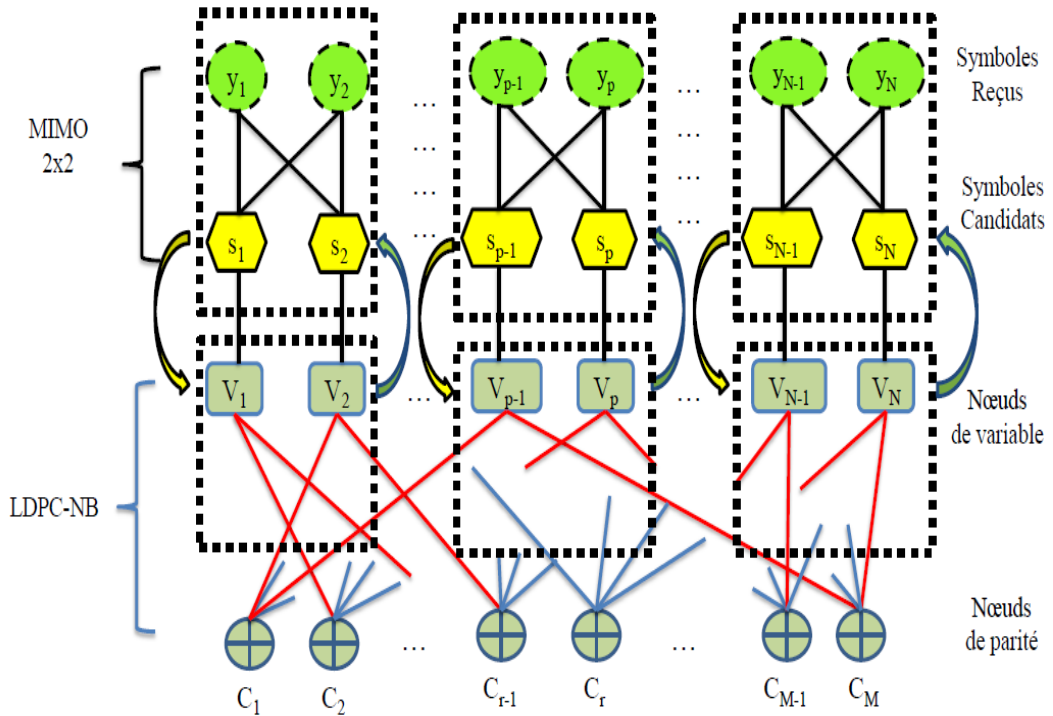


FIGURE 2.8 – Séquencement *shuffle* du GFC

favoriser la convergence [67]. Une partie des nœuds de variable peut ainsi profiter de la mise à jour des nœuds traités préalablement. Cela est possible grâce aux connexions communes via les nœuds de parité. Un autre avantage de ce type de séquençement est une diminution de la latence d'exécution pour des performances équivalentes. Le séquençement *shuffle* au sein du GFC illustré par la figure 2.8 peut être décomposé en trois étapes :

1-) Tout d'abord, nous calculons l'information *a priori* $L_{S_i \rightarrow y_j}^{(t)}$ du ($i^{\text{ème}}$) symbole candidat S_i vers le ($j^{\text{ème}}$) symbol reçu connecté y_j via l'équation (2.18) :

$$L_{S_i \rightarrow y_j}^{(t)} = \sum_{\substack{k=1 \\ k \neq j}}^{N_r} L_{y_k \rightarrow S_i}^{(t)} + L_{pr \rightarrow S_i}^{(t-1)} = L_{S_i}^{(t)} - L_{y_j \rightarrow S_i}^{(t-1)}$$

2-) Dans le nœud y_j , le vecteur LRV de chaque symbole candidat S_i est calculé. Durant la ($t^{\text{ème}}$) itération, le message envoyé du ($j^{\text{ème}}$) symbole reçu y_j vers le ($i^{\text{ème}}$) symbole candidat connecté S_i peut être exprimée par l' équation (2.10) :

$$L_{y_j \rightarrow S_i}^{(t)} = [L_{[S_i, \alpha_0]}; L_{[S_i, \alpha_1]}; \dots L_{[S_i, \alpha_{63}]}]^T$$

Et l'équation (2.16) :

$$L_{[S_i, \alpha_k]} \approx \max_{\substack{S_i = \alpha_k \\ S_m \in GF(64), (m \neq i)}} \left\{ \mu + \sum_{\substack{(m=1|m \neq i) \\ \alpha_k \in GF(64)}}^{N_t} L_{[S_m, \alpha_k]} \right\} \\ - \max_{\substack{S_i = \alpha_0 \\ S_m \in GF(64), (m \neq i)}} \left\{ \mu + \sum_{\substack{(m=1|m \neq i) \\ \alpha_k \in GF(64)}}^{N_t} L_{[S_m, \alpha_k]} \right\}$$

3-) Ensuite, la mise à jour du symbole candidat S_i est obtenue via l'équation (2.19) :

$$L_{S_i}^{(t)} = \sum_{k=1}^{N_r} L_{y_k \rightarrow S_i}^{(t)} + L_{pr \rightarrow S_i}^{(t-1)}$$

Durant la $t^{\text{ème}}$ itération, le message transmis du symbol reçu S_i vers le nœud de variable connecté V_p au sein du décodeur LDPC-NB peut être exprimée par l'équation (2.20) :

$$L_{pr \rightarrow V_p}^{(t)} = L_{S_i}^{(t)} - L_{pr \rightarrow S_i}^{(t-1)}$$

Le décodeur LDPC-NB répète que les étapes citées dans la section 2.3.3 uniquement pour les couples de nœuds variables reçus par le détecteur. Après le décodage d'un couple, les informations extrinsèques sont renvoyées vers le détecteur afin d'améliorer la détection du couple suivant. Les messages provenant du décodeur sont exploités comme des informations *a priori* lors du nouveau traitement. Ce processus est répété jusque la détection et le décodage de l'ensemble de la trame reçue et peut être itéré par plusieurs âssages successifs sur la même trame.

2.5 Analyse du processus itératif du récepteur à l'aide de diagrammes EXIT

Les performances asymptotiques de systèmes de réception itératifs sur des canaux à évanouissements théoriques dépendent de l'ordre de diversité du système de transmission. Des traitements itératifs à la réception ne convergent vers ces performances asymptotiques que pour un rapport signal à bruit élevé. En fait, le seuil de convergence dépend des paramètres du système, tel que l'algorithme de traitement utilisé. Ce seuil de convergence est un paramètre important pour la comparaison de systèmes de réception. Une méthode d'analyse consiste à étudier la convergence de systèmes itératifs pouvant se produire au-delà du rapport signal à bruit cible.

La méthode des diagrammes EXIT (*EXtrinsic Information Transfert*) a été introduite par Ten Brink en 2001 [70]. Elle permet de visualiser graphiquement les échanges d'information lors d'un traitement itératif en réception. Cette méthode consiste à représenter sur un diagramme l'évolution de l'information échangée entre un détecteur et un décodeur à entrées et sorties pondérées lors d'un traitement itératif. En clair, c'est la mesure de l'information mutuelle sortante en fonction de l'information mutuelle entrante. Cette méthode a été appliquée pour la première fois pour optimiser des algorithmes de turbo-décodage [71]. Plus tard, cet outil a été adapté aux processus itératifs au niveau réception, notamment pour la turbo-égalisation en 2002 [72] et les systèmes MIMO itératifs en 2003 [73].

2.5.1 Les diagrammes EXIT au niveau binaire

Nous noterons λ l'information extrinsèque pondérée de l'élément binaire $x \in \{-1, +1\}$ échangée entre le détecteur et le décodeur dans un système itératif. Cette information est caractérisée par la densité de probabilité conditionnelle $f(\lambda/x)$. En effet, la qualité de l'information extrinsèque est quantifiée par la mesure de l'information mutuelle moyenne $I(\lambda, x)$ apportée par λ sur x . Cette information mutuelle a pour expression :

$$I_{\lambda,x} = \frac{1}{2} \sum_{x=-1,+1} \int_{-\infty}^{+\infty} f(\lambda/x) \times \log_2 \frac{2.f(\lambda/x)}{f(\lambda/+1) + f(\lambda/-1)} d\lambda \quad (2.23)$$

où $f(\lambda/x)$ est modélisée par une variable aléatoire gaussienne de variance σ_a^2 et de moyenne $\mu_a = |E[a]|$ avec $\sigma_a^2 = 2.\mu_a$.

$f(\lambda/x)$ a pour expression :

$$f(\lambda/x) = \frac{1}{\sqrt{4\pi\mu_a}} \exp - \frac{(\lambda - \mu_a.x)^2}{4\mu_a} \quad (2.24)$$

Cette fonction suit une loi gaussienne de paramètres $N(\mu_a.x, \sigma_a^2)$. Notons également que la densité de probabilité $f(\lambda/x)$ vérifie la condition de symétrie exponentielle qui peut s'écrire sous la forme suivante :

$$f(\lambda/x) = f(\lambda/-x) \exp(\lambda) \quad (2.25)$$

De même, la symétrie suivante existe :

$$f(\lambda/x = +1) = f(-\lambda/x = -1) \quad (2.26)$$

L'expression précédente de l'information mutuelle moyenne se simplifie :

$$I_{\lambda,x} = \int_{-\infty}^{+\infty} f(\lambda/ + 1) \times \log_2 \frac{2}{1 + \exp(-\lambda)} d\lambda \quad (2.27)$$

Nous constatons que $I_{\lambda,x}$ est une fonction strictement croissante du paramètre μ_a , avec pour limite :

$$\lim_{\mu_a \rightarrow 0} I_{\lambda,x}(\mu_a) = 0, \quad \lim_{\mu_a \rightarrow +\infty} I_{\lambda,x}(\mu_a) = 1 \quad (2.28)$$

où $I_{\lambda,x} = 0$ indique que λ_x n'apporte aucune information sur x . A l'inverse, $I_{\lambda,x} = 1$ lorsque λ détermine parfaitement x .

La mesure de l'information mutuelle dépend du type d'élément à caractériser. Le protocole de mesure de l'information mutuelle entrante et sortante pour un détecteur MIMO itératif est détaillé dans [73]. Ce protocole est illustré par la figure 2.9 qui décrit le principe de la mesure de l'information mutuelle pour un détecteur opérant sur les données issues du canal et bénéficiant d'une information *a priori*. Cette mesure est basée sur deux niveaux :

1-) Le canal de transmission qui est caractérisé par la variance du bruit σ_w^2 . Ce niveau est fonction de l'énergie par bit utile E_b/N_0 (dB) du canal. Pour ce faire, le codeur de canal reçoit les bits d'informations source et rajoute de la redondance. Les bits codés par le codage canal passent par l'étape de modulation. Puis, le signal subit des perturbations en fonction de la variance du bruit σ_w^2 .

2-) La fiabilité des informations *a priori* est caractérisée par le terme σ_a^2 . L'information mutuelle entrante I_A est calculée en générant une variable aléatoire gaussienne de variance σ_a^2 et de moyenne μ_a avec $\sigma_a^2 = 2\mu_a$ suivant l'équation (2.24). A ce niveau, nous calculons les vecteurs LRV des bits codés sortants directement du codage canal en fonction du terme σ_a^2 .

L'information mutuelle sortante I_E à la sortie du détecteur est fonction de l'énergie par bit

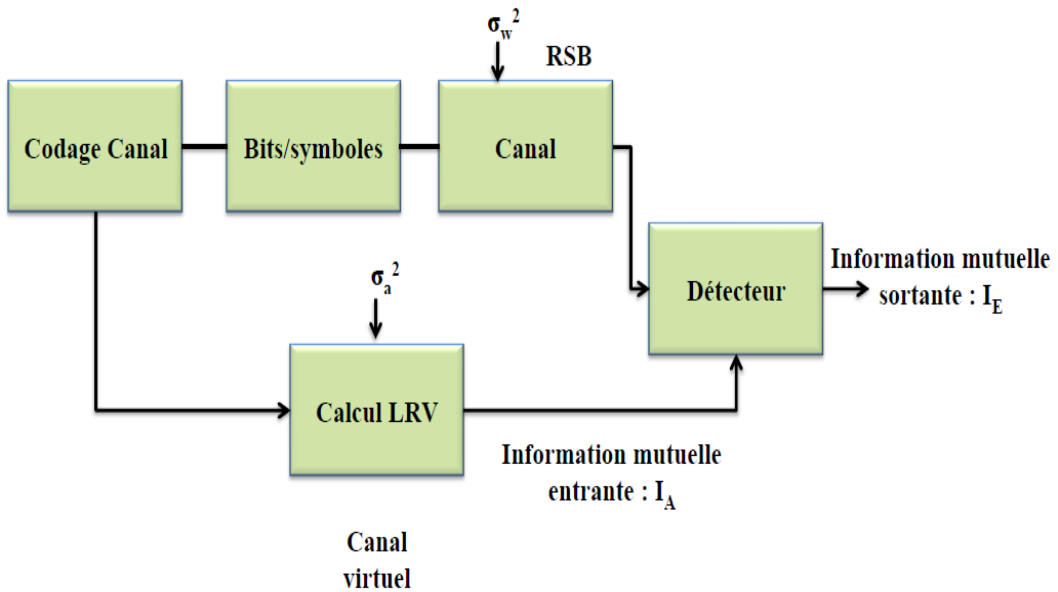


FIGURE 2.9 – Principe de mesure de l'information mutuelle entrante et sortante pour un détecteur MIMO.

utile $Eb/N0$ (dB) du canal et de l'information *a priori* I_A à l'entrée du détecteur telle que

$$I_E = T(I_A, Eb/N0) \quad (2.29)$$

Cette information est déterminée grâce à l'équation (2.23). Dans la suite du document, nous étudions les diagrammes EXIT au niveau non-binaire qui sont intéressants dans notre contexte pour le choix des paramètres du système itératif.

2.5.2 Les diagrammes EXIT au niveau non-binaire

Dans cette section, nous proposons une analyse de la convergence à l'aide de diagrammes EXIT pour des symboles non-binaires. Dans un premier temps, nous étudions cette convergence au niveau détecteur. Dans un second temps, nous étendrons l'étude au niveau récepteur itératif.

Afin de représenter l'information mutuelle décrite dans la section précédente sous une forme non-binaire pour une constellation 64 QAM, l'équation 2.23 peut s'écrire de la manière suivante :

$$I_{\lambda, \alpha_k} = \frac{1}{64} \sum_{\alpha_k=0}^{\alpha_k=63} \int_{-\infty}^{+\infty} f(\lambda/\alpha_k) \times \log_2 \frac{2 \times f(\lambda/\alpha_k)}{f(\lambda/\alpha_0) + f(\lambda/\alpha_1) + \dots + f(\lambda/\alpha_{63})} d\lambda \quad (2.30)$$

où $f(\lambda/\alpha_k)$ désigne la densité de probabilité non-binaire que λ soit égale à α_k . Cette densité est représentée par la somme de densités de probabilité binaires générées selon une distribution Gaussienne pour Q bits représentant un symbole non-binaire. Grâce à cette équation, nous pouvons calculer l'information mutuelle non-binaire du système proposé.

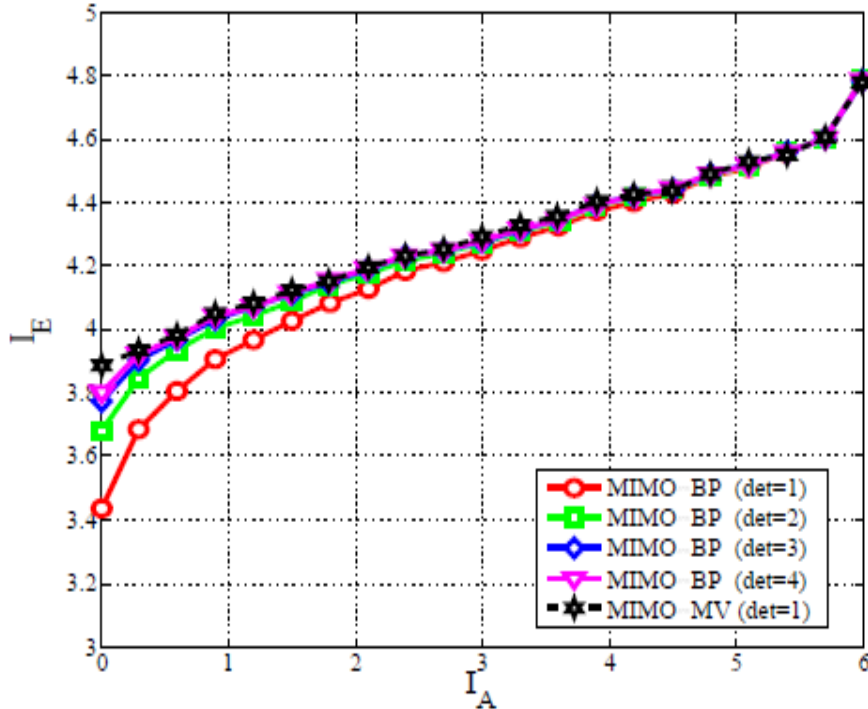


FIGURE 2.10 – Trajectoires des détecteurs MIMO pour un rapport RSB=11dB

Au niveau détecteur, nous mesurons l'information mutuelle à la sortie du détecteur en fonction de l'information *a priori* à l'entrée de ce dernier et le rapport RSB comme illustré dans la figure 2.9. Notons que dans les figures suivantes, NDD désigne le nombre d'inter-itérations entre le détecteur et le décodeur. det et dec indiquent respectivement le nombre d'itérations au sein du détecteur et du décodeur. En vue de comparer les performances du détecteur MIMO basé sur la propagation de croyance (BP), le détecteur MIMO basé sur l'algorithme de maximum de vraisemblance (MV) est retenu comme référence.

Dans la figure 2.10, I_A représente l'information mutuelle à l'entrée du détecteur et I_E représente l'information mutuelle à sa sortie. Cette figure montre que la trajectoire du détecteur MIMO basé sur l'algorithme de propagation de croyance converge vers celle du détecteur MIMO basé sur une détection optimale MV après seulement 4 intra-itérations au sein du détecteur.

Au niveau réception, le décodeur de canal reçoit des informations provenant du détecteur. La mesure de l'information mutuelle est donnée dans la figure 2.11. Dans ce cas, la sortie du décodeur ne dépend que de l'information mutuelle sortante du détecteur MIMO et de l'information mutuelle passant du $r^{\text{ème}}$ nœud de parité au $p^{\text{ème}}$ nœud de variable. Dans cette figure, I_A représente l'information mutuelle à l'entrée du détecteur et désigne également l'information mutuelle à la sortie du décodeur lors de l'itération précédente [65]. De même, I_E représente l'information mutuelle à la sortie du détecteur. Cette information est fonction du rapport signal à bruit des données issues du canal et de l'information *a priori* I_A provenant du décodeur lors de l'itération précédente. I_E représente également l'information mutuelle à l'entrée du décodeur LDPC-NB.

Au sein du décodeur, I_{VversP} représente l'information mutuelle passant du $p^{\text{ème}}$ nœud de variable au $r^{\text{ème}}$ nœud de parité. Alors que I_{PversV} représente l'information mutuelle passant du $r^{\text{ème}}$ nœud de parité au $p^{\text{ème}}$ nœud de variable. L'information mutuelle I_{VversP} est donnée par :

$$I_{VversP}^{(i)} = T(I_{PversV}^{(i)}, I_E) \quad (2.31)$$

avec $I_{PversV}^{(0)} = 0$, $I_{VversP}^{(0)} = I_E$ et i designant le nombre d'itérations au sein du décodeur. L'information mutuelle I_{PversV} correspond à l'état du décodage du nœud de parité tel que :

$$I_{PversV}^{(i)} = T(I_{VversP}^{(i-1)}) \quad (2.32)$$

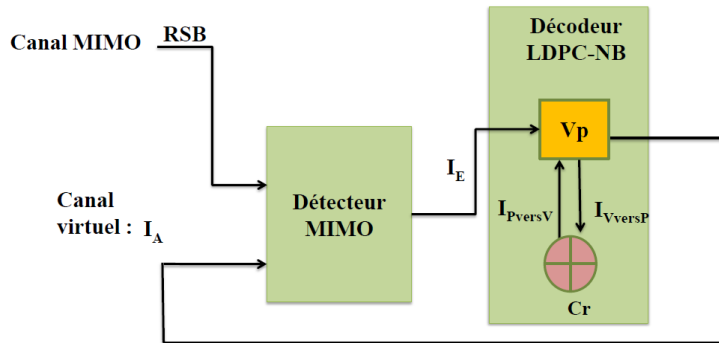


FIGURE 2.11 – Mesure de l'information mutuelle entrante et sortante pour un récepteur itératif.

Les trajectoires des diagrammes EXIT qui sont obtenues dans [74] sont appliquées dans cette section. Ainsi, les trajectoires du récepteur MIMO basé sur la propagation de croyance peuvent être divisées en plusieurs segments (verticales et horizontales) qui relient respectivement les informations mutuelles au niveau des sorties du détecteur et du décodeur. Dans la suite, nous allons étudier les trajectoires de la convergence à l'aide des diagrammes EXIT en appliquant deux types de séquençement *flooding* [65], [66] et *shuffle* [67].

Lors de l'application d'un séquençement *flooding*, la trajectoire du récepteur MIMO basé sur l'algorithme de propagation de croyance peut être divisée en 16 segments qui relient les points de 0 à 16 comme illustré par la figure 2.12. Les points impairs et pairs représentent respectivement l'information mutuelle au niveau des sorties du détecteur et du décodeur. Le premier point de la trajectoire MIMO basé sur l'algorithme de propagation de croyance est l'état initial du décodeur après 4 intra-itérations au sein du détecteur (ligne verticale).

Le second point correspond à l'état du décodeur après une intra-itération de décodage au sein du décodeur LDPC-NB. Le résultat du décodage est exploité comme une nouvelle information mutuelle à l'entrée du détecteur. Ce processus se poursuit jusqu'au point (6,6). Notons que la courbe rouge *PversV* correspond à l'état du décodage au sein du décodeur LDPC-NB. Il apparaît sur la figure 2.12 que la trajectoire du détecteur MIMO basé sur l'algorithme de propagation de croyance (courbe bleue) converge vers celle du détecteur MIMO basé sur une détection optimale MV (courbe verte) selon un ordonnancement pertinent pour les inter-itérations au sein du récepteur itératif.

Cet ordonnancement permet une convergence rapide vers des performances ayant un moindre coût calculatoire. Il implique 8 inter-itérations entre le détecteur et le décodeur, 4 intra-itérations

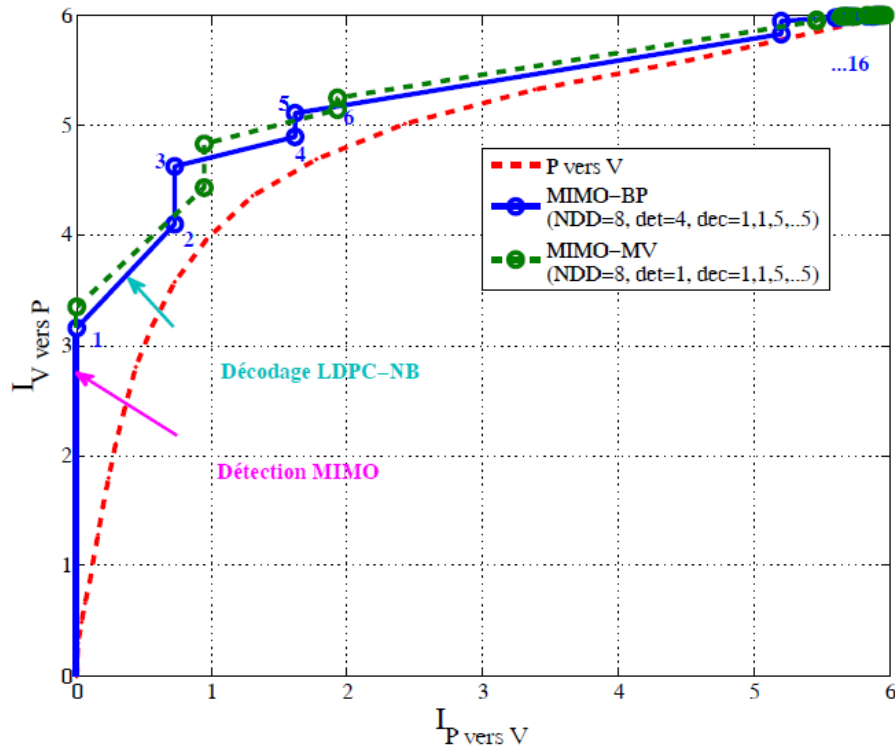


FIGURE 2.12 – Trajectoires des récepteurs MIMO pour un séquençement *flooding* pour un RSB=11 dB

au sein du détecteur. Au niveau décodeur, 1 intra-itération est nécessaire durant les deux premières inter-itérations, puis 5 intra-itérations par la suite.

La figure 2.13 illustre les trajectoires du récepteur MIMO basé sur l'algorithme de propagation de croyance pour différents séquencements (*flooding* ou *shuffle*). Notons que les points intermédiaires à la sortie du détecteur MIMO qui sont indiqués sur la figure 2.12 ne sont pas représentés dans cette figure. Par exemple, pour obtenir un point particulier dans le diagramme EXIT tels que (5.2, 5.8), en cas d'application du séquencement *flooding* avec $[NDD = 8; (det, dec) = [(4, 1), (4, 1), (4, 5), (4, 5), (4, 5), (4, 5), (4, 5), (4, 5)]]$, nous avons besoin de 4 inter-itérations globales entre le détecteur et le décodeur.

Cela aboutit au final à 16 intra-itérations au sein du détecteur MIMO basé sur l'algorithme de propagation de croyance et à 12 intra-itérations au sein du décodeur LDPC-NB. Ce nombre d'itérations peut être encore réduit à l'aide d'un séquencement *shuffle* avec $[NDD = 32; (det, dec) = [(1, 1), (1, 1), (1, 1), \dots, (1, 1)]]$. En effet, ce dernier nécessite que 5 inter-itérations globales entre le détecteur et le décodeur. Cela donne 5 intra-itérations au sein du détecteur MIMO basé sur l'algorithme de propagation de croyance et 5 intra-itération au sein du décodeur LDPC-NB.

Nous remarquons que pour des performances équivalentes, l'application d'un séquencement *shuffle* au sein d'un récepteur itératif offre un meilleur compromis au niveau de la complexité calculatoire et de la latence.

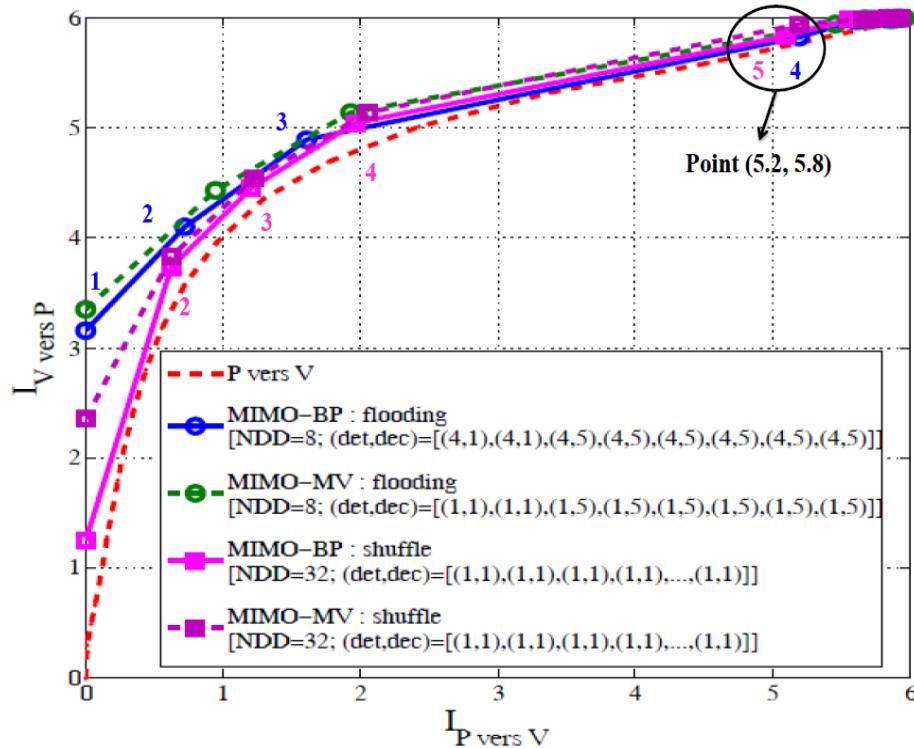


FIGURE 2.13 – Trajectoires des récepteurs MIMO pour différents séquencements avec un RSB=11 dB

2.6 Etudes expérimentales

Cette dernière section détaille des résultats de simulation qui confirment les choix faits à l'issue de l'étude des diagrammes EXIT. Dans la suite, nous allons présenter les performances de notre système de réception puis comparer ses performances selon différents paramètres. Les performances sont mesurées en terme de taux d'erreur symbole (TES) en fonction du rapport signal à bruit (RSB). Afin d'évaluer les performances du détecteur sous-optimal MIMO-BP, un détecteur optimal MIMO-MV basé sur les mêmes paramètres est utilisé comme référence. Les résultats de simulations illustrés par les figures suivantes ont été obtenus pour un système MIMO associé à une modulation 64 QAM suivie d'un multiplexage spatial 2x2.

2.6.1 Performances du séquençement *flooding* : détection bit par bit versus détection symbole par symbole

Nous allons représenter dans cette sous section une comparaison au niveau des performances entre une détection bit par bit et une détection symbole par symbole basées sur l'algorithme de propagation de croyance dans un contexte itératif et non-itératif. Les comparaisons comprendront également les performances d'une détection optimale basée sur l'algorithme MV pour un même jeu de paramètre afin d'évaluer notre système par rapport à un système idéal. Le gain obtenu à l'aide du traitement itératif au sein d'un récepteur MIMO peut être observé dans les figures 2.14 et 2.15.

Notons qu'afin d'atteindre les meilleures performances, les courbes sont tracées pour un nombre de 8 intra-itérations au sein du décodeur LDPC-NB ($dec = 8$), 4 intra-itérations au sein du détecteur MIMO ($det = 4$) et 8 inter-itérations ($NDD = 8$) globales entre le détecteur et le décodeur au sein du récepteur itératif. Ainsi pour chaque inter-itération globale du récepteur, 8 intra-itérations au sein du décodeur LDPC-NB et 4 intra-itérations au sein du détecteur MIMO basé

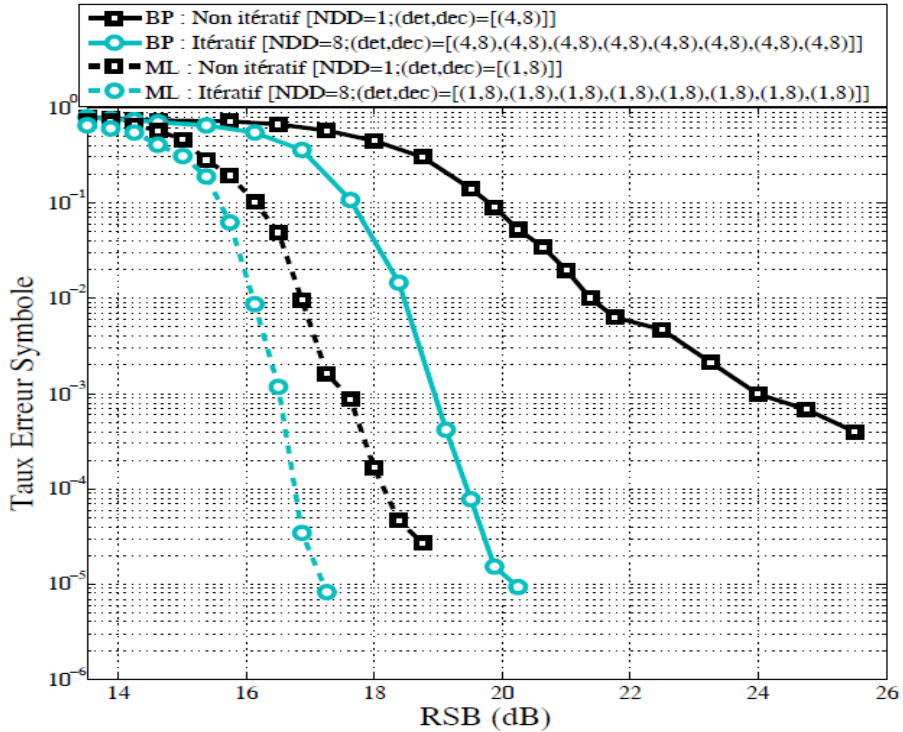


FIGURE 2.14 – Performances en termes TES d'une détection bit par bit

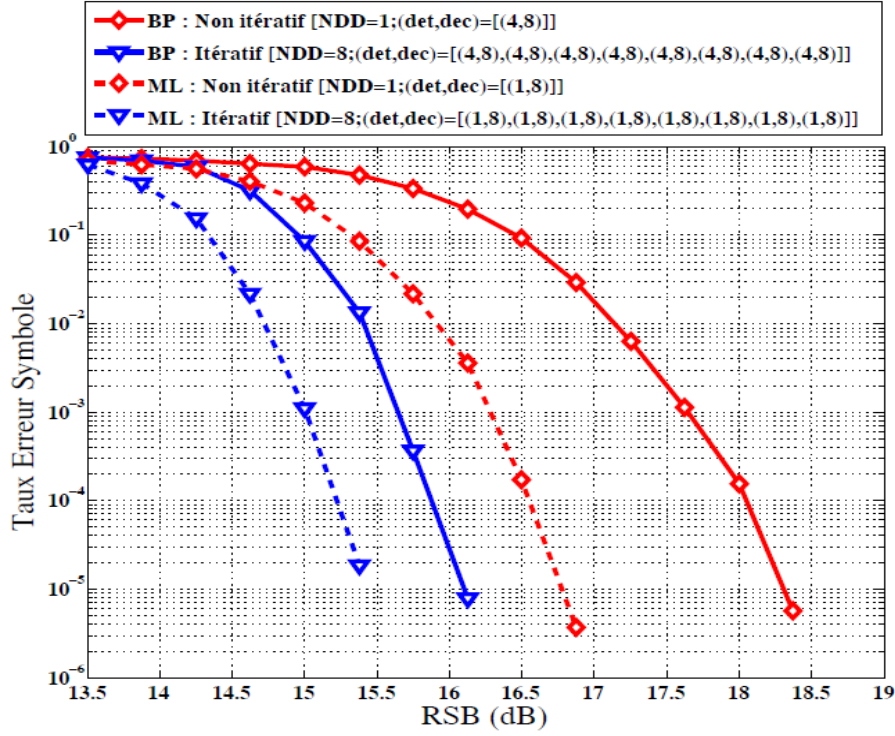


FIGURE 2.15 – Performances en termes TES d'une détection symbole par symbole

sur la propagation de croyance sont effectuées. Par conséquent, le système nécessite 64 intra-itérations de décodage LDPC-NB et 32 intra-itérations de détection MIMO basé sur la propagation de croyance.

Des comparaisons sont faites pour un TES de 10^{-4} . Dans le cas d'une détection bit par bit, un code LDPC binaire adopté dans la norme *IEEE WiMAX* [60] a été utilisé comme référence. Les caractéristiques de ce code LDPC sont similaires à celles du code LDPC-NB avec une longueur de trame $N = 2304$ bits et un rendement de codage $R = 1/2$.

Les figures 2.14 et 2.15 donnent respectivement les performances d'une détection sous-optimale bit par bit et d'une détection sous-optimale symbole par symbole dans un contexte itératif et non-itératif. Dans la figure 2.14, nous remarquons que dans le cas d'un traitement non-itératif, l'application d'une détection MIMO basée sur l'algorithme BP introduit une pénalité de plus de 7,0 dB par rapport à une détection optimale MV.

Lorsque le traitement itératif est introduit, cette pénalité est réduite à 2,8 dB. De même dans la figure 2.15 où la détection est faite symbole par symbole, ($\log_2 M$ bits représentent un symbole non-binaire), l'application d'une détection non-itérative sous optimale basée sur l'algorithme BP introduit une pénalité de plus de 1,5 dB par rapport à une détection optimale basée sur l'algorithme MV. Cependant, lorsque le traitement itératif est introduit, cette pénalité est réduite à moins de 0,5 dB. Le récepteur proposé surpasse le récepteur optimal non-itératif de 0,5 dB.

Par conséquent, pour une détection symbole par symbole itérative, le récepteur sous optimal proposé se rapproche des performances du récepteur optimal en favorisant une forte diminution de la complexité calculatoire et de la latence.

2.6.2 Performances du séquençement *flooding* retenu à l'aide des diagrammes Exit au niveau non-binaire.

Nous avons vu dans la sous section précédente que les meilleures performances de notre système sont obtenues par le récepteur avec un séquençement *flooding* bénéficiant de 8 inter-itérations au sein du récepteur itératif. De plus, pour chaque inter-itération, 8 intra-itérations sont nécessaires au sein du décodeur LDPC-NB et 4 intra-itérations au sein du détecteur MIMO. Dans cette sous section, nous allons donner les performances du système qui permet de réduire le nombre des intra-itérations au sein du décodeur LDPC-NB.

Pour ce faire, l'ordonnancement pertinent trouvé à l'issue de l'étude des diagrammes EXIT est appliqué. Ces performances sont illustrées dans la figure 2.16. Dans ce cas, aucune dégradation significative au niveau des performances est observée après l'application de l'ordonnancement proposé. Par conséquent, une réduction d'environ 50% du nombre d'intra-itérations au sein du décodeur LDPC-NB est obtenue sans pénalité au niveau des performances. Cette réduction conduit à diminuer de moitié la complexité calculatoire et la latence au sein du décodeur LDPC-NB. Ce qui est un résultat probant.

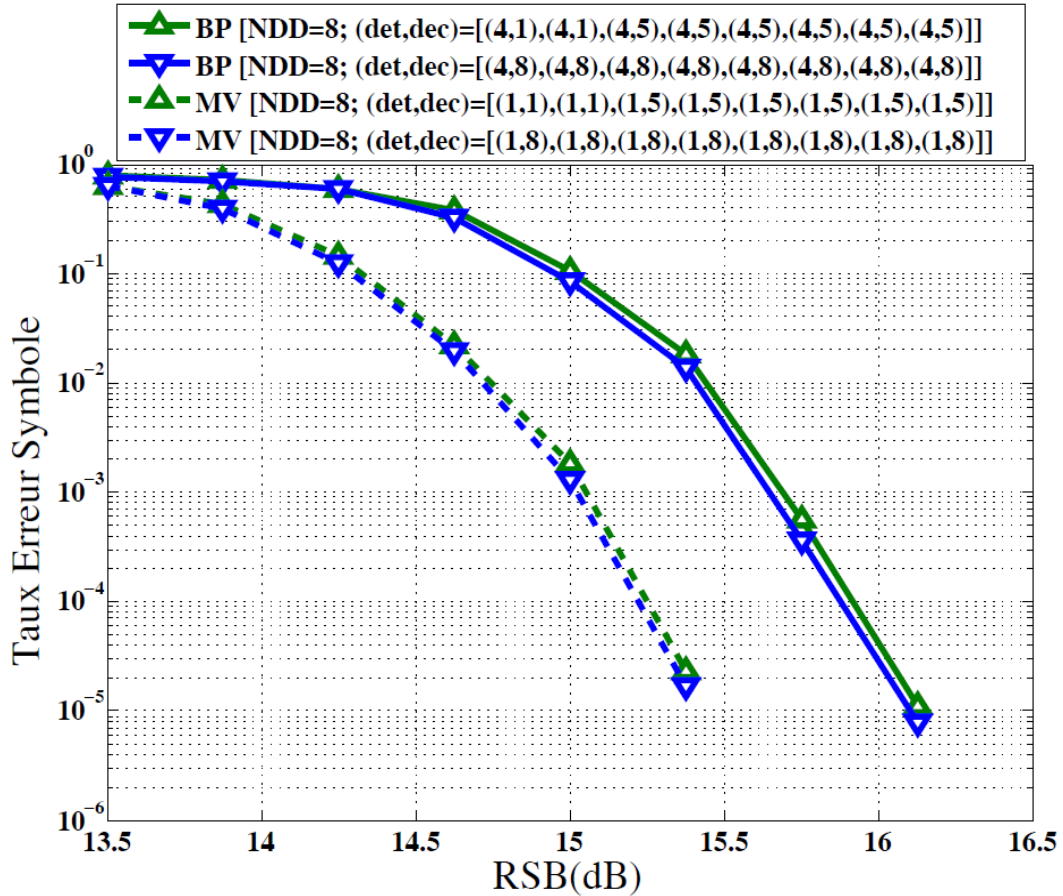


FIGURE 2.16 – Performances en termes TES de l'ordonnancement retenu à l'aide des diagrammes EXIT.

2.6.3 Performances du séquençement *shuffle* versus séquençement *flooding*

La figure 2.17 représente les performances au niveau TES du système proposé en appliquant, cette fois, un séquençement *shuffle* au sein du récepteur itératif. Nous avons vu lors de l'étude des diagrammes EXIT dans la sous section (2.5.2) que les meilleures performances de notre système sont obtenues par le récepteur avec un séquençement *shuffle* bénéficiant de 32 inter-itérations au sein du récepteur itératif.

Pour chaque inter-itération, 1 intra-itération est nécessaire au sein du décodeur LDPC-NB et 1 intra-itération au sein du détecteur MIMO. Ce système nécessite au total 32 intra-itérations au sein du détecteur MIMO et de 32 intra-itérations au sein du décodeur LDPC-NB. C'est équivalent aux nombres d'intra-itérations pour un séquençement *flooding* au sein du récepteur itératif. Nous allons comparer les performances du système proposé en conservant donc la même complexité au niveau du nombre d'intra-itérations entre les différents séquençements.

Dans la figure 2.17, nous remarquons que le gain de l'application d'un séquençement *shuffle* avec un détecteur basé sur l'algorithme MV est négligeable par rapport à l'application d'un séquençement *flooding*. Toutefois, dans le cas d'un détecteur basé sur l'algorithme BP, un gain de 0,4 dB est observée lors de l'application d'un séquençement *shuffle* par rapport à l'application d'un séquençement *flooding*. En plus des gains de performances observés, le séquençement *shuffle* réduit considérablement la latence du récepteur itératif. Cette approche s'avère donc particulièrement adaptée pour une mise en œuvre matérielle. Par conséquent, la proposition d'un traitement itératif

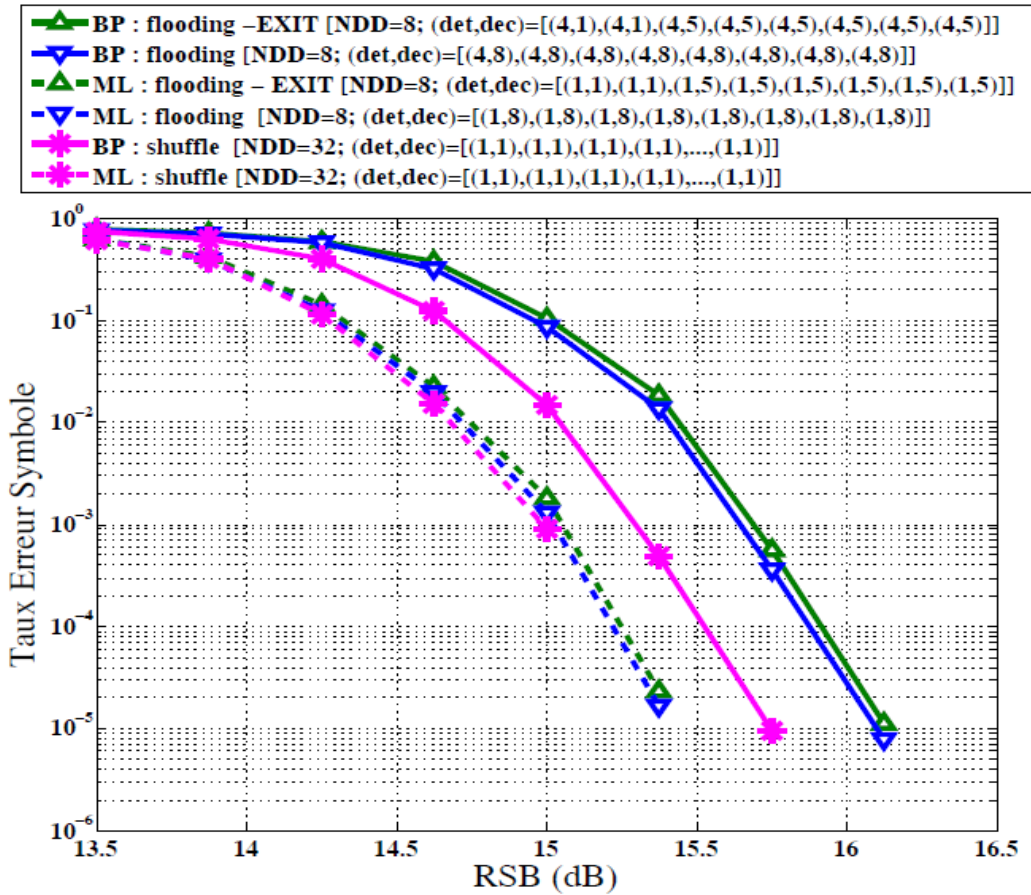


FIGURE 2.17 – Performances du système de réception proposé selon différents séquençements

en réception basé sur les graphes respectifs du détecteur MIMO et du décodeur LDPC-NB selon un séquençement *shuffle* doit permettre la conception de récepteurs performants et peu complexes ayant une faible latence dans un contexte itératif.

Cette étude effectuée pour différents algorithmes de détection et selon différents séquençements au niveau du récepteur itératif a fait l'objet de plusieurs publications [75],[76].

2.7 Conclusion

Dans ce chapitre, nous avons étudié la combinaison d'un détecteur MIMO basé sur la détection symbole par symbole suivie d'un décodeur du code LDPC-NB. Cette étude a été effectuée pour différents algorithmes de détection et selon différents séquençements au niveau du récepteur itératif.

Tout d'abord, nous avons montré que le traitement symbole par symbole améliore considérablement les performances du système non-itératif par rapport à un traitement bit par bit. En outre, la pénalité d'introduction d'une détection sous-optimale basée sur l'algorithme BP est fortement réduite lorsque le traitement itératif est appliquée entre le détecteur et le décodeur.

Ensuite, des analyses graphiques à l'aide de diagrammes EXIT ont été faites pour trouver l'ordonnancement pertinent au niveau des intra-itérations et des inter-itérations au sein du récepteur itératif MIMO.

Une représentation graphique du graphe à facteur conjoint (GFC) du récepteur MIMO avec un multiplexage spatial suivi d'un décodage de code LDPC-NB permet d'explorer un séquençement *shuffle* lors de la détection et du décodage. L'intérêt majeur de ce séquençement est de favoriser la convergence en diminuant la latence d'exécution pour des performances équivalentes.

Algorithme de détection et de décodage conjoints à faible complexité calculatoire

La recherche portant sur des traitements numériques associés à la technologie MIMO s'est focalisée dans un premier temps sur les aspects théoriques. De nombreux travaux ont notamment porté sur l'établissement des performances théoriques de tels systèmes et sur la proposition d'ensembles émetteur/récepteur permettant de tirer bénéfice de l'utilisation de plusieurs antennes. Malgré la multitude des travaux de recherche dans ce domaine, la complexité du détecteur MIMO et la latence associée demeurent parmi les principaux axes d'amélioration possibles.

Les techniques de détection MIMO basées sur la propagation de croyance, que nous avons développées dans le chapitre 2, apparaissent comme une solution algorithmique offrant de forts potentiels pour une implémentation matérielle. La proposition d'un traitement itératif en réception basé sur des graphes bipartites pour le détecteur MIMO et le décodeur de code LDPC non-binaire doit favoriser la conception de récepteurs flexibles ayant une faible latence dans un contexte itératif.

Dans ce chapitre, nous allons décrire une proposition d'algorithme de détection et de décodage conjoints ayant une complexité calculatoire réduite. Nous allons diminuer la complexité de notre récepteur itératif à deux niveaux. Dans un premier temps, nous réduirons le coût du calcul initial des distances euclidiennes (D_{Euc}) pour le détecteur MIMO. Puis, nous aborderons la réduction du nombre d'échanges d'informations au cours de chaque processus itératif.

3.1 Calcul récursif des distances euclidiennes par exploitation des points voisins de la constellation

Avant de présenter l'algorithme de détection à faible complexité, nous allons décrire une nouvelle méthode de calcul du rapport de vraisemblance (LRV) de tous les points d'une constellation par récursion à partir d'un seul point choisi.

Comme nous l'avons mentionné dans le chapitre précédent, nous modélisons une chaîne de transmission MIMO intégrant un codeur LDPC non-binaire sur $GF(64)$ avec une modulation d'amplitude en quadrature 64-QAM suivie d'un multiplexage spatial composé de $N_t = 2$ antennes à l'émission et de $N_r = 2$ antennes à la réception. Notre algorithme de détection basé sur la propagation de croyance calcule le vecteur des informations souples du rapport de vraisemblance (LRV) des symboles candidats S_i devant être transmis au décodeur LDPC non-binaire.

Rappelons que l'expression du vecteur LRV d'un symbole candidat S_i a été explicitée dans

le chapitre 2 :

$$L_{S_i} = [L_{[S_i, \alpha_0]}; L_{[S_i, \alpha_1]}; \dots L_{[S_i, \alpha_{63}]}]^T \quad (3.1)$$

où $L_{[S_i, \alpha_k]}$, $k = 0, 1, 2, \dots, 63$, représente la fiabilité pour que le symbole candidat S_i soit égal à $\alpha_k \in GF(64)$ tel que :

$$L_{[S_i, \alpha_k]} \approx \max_{\substack{S_i = \alpha_k \\ S_m \in GF(64), (m \neq i)}} \left\{ \mu + \sum_{\substack{(m=1|m \neq i) \\ \alpha_k \in GF(64)}}^{N_t} L_{[S_m, \alpha_k]} \right\} - \max_{\substack{S_i = \alpha_0 \\ S_m \in GF(64), (m \neq i)}} \left\{ \mu + \sum_{\substack{(m=1|m \neq i) \\ \alpha_k \in GF(64)}}^{N_t} L_{[S_m, \alpha_k]} \right\} \quad (3.2)$$

où $\mu = -\frac{1}{\sigma_w^2} |y_j - \sum_i h_{j,i} S_i|^2$ avec σ_w^2 la variance du bruit additif blanc Gaussien, $j = 1, 2, \dots, N_r$ et $i = 1, 2, \dots, N_t$.

Dans cette section et dans la section 3.2, nous détaillerons la simplification du calcul du terme $\min|\mu|$. Puis, dans la section 3.3, nous allons nous attacher à simplifier l'échange des messages s'exprimant par le terme $\sum_{\substack{(m=1|m \neq i) \\ \alpha_k \in GF(64)}}^{N_t} L_{[S_m, \alpha_k]}$ dans un contexte itératif.

Il faut noter qu'une modulation d'amplitude en quadrature M -QAM se compose des points suivants :

$$S_i = \{s = s_I + js_Q | s_I, s_Q \in A_c\} \quad (3.3)$$

où A_c se définit comme suit :

$$A_c = \left\{ \frac{1}{\sigma_s} (-\sqrt{M} + 1 + 2P) | P = 0, 1, \dots, \sqrt{M} - 1 \right\} \quad (3.4)$$

avec σ_s la facteur de normalisation. Par exemple, dans notre étude où la constellation étudiée est d'ordre 64-QAM, σ_s est égal à $\sqrt{42}$. Il est à noter que $h_{j,i}^I$ (resp. $h_{j,i}^Q$) représente le coefficient du canal non sélectif en fréquence avec des évanouissements Rayleigh de la $i^{\text{ème}}$ antenne d'émission à la $j^{\text{ème}}$ antenne de réception sur l'axe I (resp. Q) de la constellation avec $j = 1, 2, \dots, N_r$ et $i = 1, 2, \dots, N_t$. Dans ce contexte, le symbol reçu par la première antenne de réception pour un canal ayant un multiplexage spatial ($N_t = N_r = 2$) a pour expression :

$$y_1 = \begin{pmatrix} y_1^{(I)} \\ y_1^{(Q)} \end{pmatrix} = \begin{pmatrix} h_{1,1}^{(I)} & -h_{1,1}^{(Q)} & h_{1,2}^{(I)} & -h_{1,2}^{(Q)} \\ h_{1,1}^{(Q)} & h_{1,1}^{(I)} & h_{1,2}^{(Q)} & h_{1,2}^{(I)} \end{pmatrix} \begin{pmatrix} s_1^{(I)} \\ s_1^{(Q)} \\ s_2^{(I)} \\ s_2^{(Q)} \end{pmatrix} + \begin{pmatrix} n_1^{(I)} \\ n_1^{(Q)} \end{pmatrix} \quad (3.5)$$

	Multiplication Réelle (MR)	Addition Réelle (AR)
$1 D_{Euc}(S_1, S_2)$	10	9
$64^2 D_{Euc}(S_1, S_2)$	(10×64^2) = 40960	(9×64^2) = 36864

TABLE 3.1 – Coût calculatoire du terme $\min |\mu|$ pour un symbole reçu.

où $n_1 = n_1^{(I)} + j n_1^{(Q)}$ représente le bruit additif blanc Gaussien associé à la première antenne de réception.

Par conséquent, le calcul du vecteur des informations souples du rapport de vraisemblance d'une constellation nécessite le calcul des distances euclidiennes entre le symbole reçu à la sortie du canal et tous les points de la constellation. Pour des constellations d'ordre élevé tels que la 64-QAM et avec deux antennes à l'émission $N_t = 2$, la complexité du calcul de ces distances euclidiennes s'avère très coûteux. En effet, le calcul du terme $\min \{ |y_j - \sum_i h_{j,i} S_i|^2 \}$ requiert $\{64 \times 64\}$ distances euclidiennes. Or, le calcul de chaque distance euclidienne nécessite 10 Multiplications Réelles (MR), 9 Additions Réelles (AR) / Soustractions Réelles (SR). Il est à noter qu'une soustraction est aussi une addition avec des nombres entiers lorsqu'ils sont représentés en complément à 2. Le coût calculatoire du terme $\min \{ |y_j - \sum_i h_{j,i} S_i|^2 \}$ qui désigne la D_{EUC} entre un symbol reçu et deux symboles candidats S_1 et S_2 est estimé dans le tableau 3.1. Parmi ces différents types d'opérations, il est primordial de diminuer le nombre de multiplications.

Afin de réduire la complexité du calcul de $\{ |y_j - \sum_i h_{j,i} S_i|^2 \}$, nous proposons de calculer toutes les distances euclidiennes des symboles candidats par récursions successives. Considérons que ce point de départ dans la constellation est $\{I = -7.0; Q = -7.0\}$ pour les deux symboles candidats S_1 et S_2 :

$$\{s_1 = (s_1^{(I)}; s_1^{(Q)}) = (-7.0; -7.0) | s_2 = (s_2^{(I)}; s_2^{(Q)}) = (-7.0; -7.0)\}$$

La distance euclidienne calculée à partir de ce point de départ pour les deux symboles candidats a pour expression :

$$D_{EUC}(S_1, S_2) = \{s_1 = (-7.0, -7.0) | s_2 = (-7.0, -7.0)\} |D_{EUC}^I|^2 + |D_{EUC}^Q|^2 \quad (3.6)$$

tel que

$$\begin{aligned} D_{EUC}^I &= (y_1^{(I)} - (-7.0 \times h_{1,1}^{(I)}) - (-7.0 \times h_{1,1}^{(Q)}) + (-7.0 \times h_{1,2}^{(I)}) - (-7.0 \times h_{1,2}^{(Q)})) \\ D_{EUC}^Q &= (y_1^{(Q)} - (-7.0 \times h_{1,1}^{(Q)}) + (-7.0 \times h_{1,1}^{(I)}) + (-7.0 \times h_{1,2}^{(Q)}) + (-7.0 \times h_{1,2}^{(I)})) \end{aligned}$$

Le calcul de la distance euclidienne de ce point nécessite 10 MR, 9 AR. A partir de l'équation (3.6), il est possible de déduire la distance euclidienne d'un point voisin au point de départ par une simple addition du terme $\{2 \times h_{j,i}\}$. Par exemple, considérons l'expression de la distance euclidienne de ce point :

$$\{s_1 = (s_1^{(I)}; s_1^{(Q)}) = (-7.0; -5.0) | s_2 = (s_2^{(I)}; s_2^{(Q)}) = (-7.0; -7.0)\}$$

Nous pouvons additionner le terme $\{2 \times h_{1,1}^{(Q)}\}$ à la distance euclidienne précédente. Cette approche ne nécessite qu'une seule addition. Puis, une seule multiplication est nécessaire pour le carré du terme

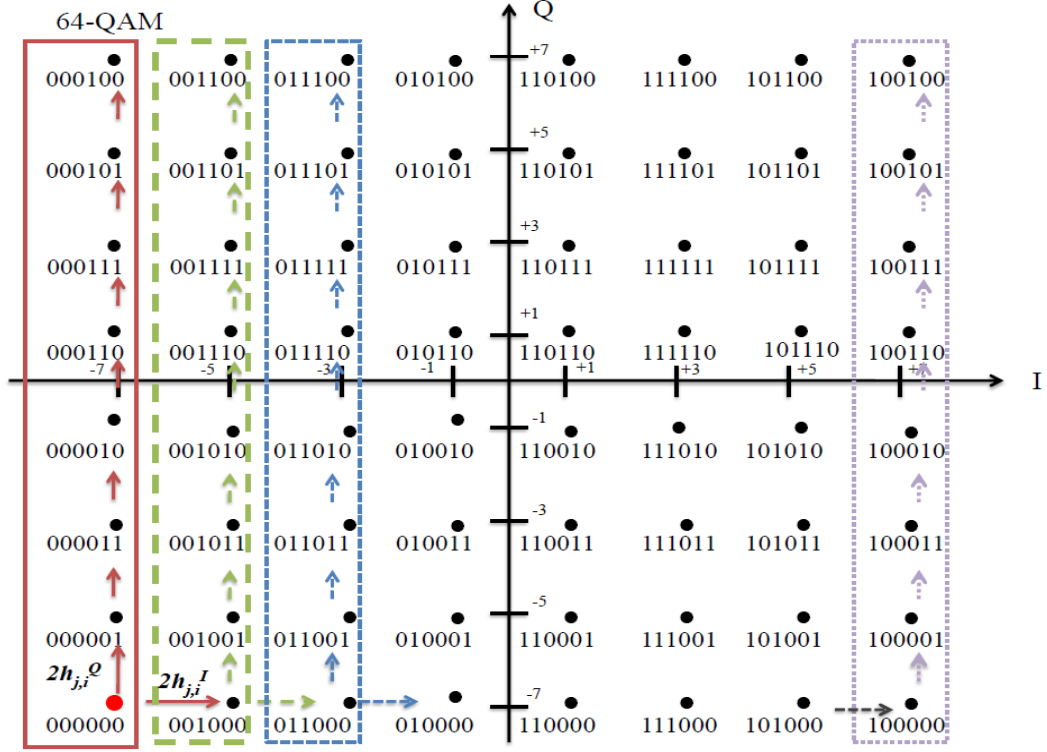


FIGURE 3.1 – Illustration du calcul récursif des distances euclidiennes par exploitation des points voisins dans le cas d’une constellation 64-QAM.

obtenu $|D_{EUC}^Q|^2$.

Pour une constellation 64-QAM, nous divisons notre constellation en 8 colonnes selon l’axe I avec $\{I=-7.0, -5.0, \dots, +7.0\}$. Chaque colonne est composée de 8 points selon l’axe Q avec $\{Q=-7.0, -5.0, \dots, +7.0\}$ comme illustré dans la figure 3.1.

La méthode proposée se décompose en 5 étapes :

1-) Calculer les termes suivants en appliquant un décalage à gauche :

$$\{(2 \times h_{1,1}^{(I)}), (2 \times h_{1,1}^{(Q)}), (2 \times h_{1,2}^{(I)}), (2 \times h_{1,2}^{(Q)})\}$$

.

2-) En appliquant l’équation (3.6), calculer la distance euclidienne du point de départ

$$\{s_1 = (s_1^{(I)}; s_1^{(Q)}) = (-7.0; -7.0) | s_2 = (s_2^{(I)}; s_2^{(Q)}) = (-7.0; -7.0)\}$$

Cette étape requiert seulement 10 MR, 9 AR.

3-) En déduire les distances euclidiennes des 7 autres points de la première colonne du symbole candidat S_1 en fixant le deuxième symbole candidat S_2 . Durant chaque décalage, on effectue une simple addition à la valeur du point précédent par le terme $2 \times h_{j,i}^{(Q)}$ et ensuite une seule multiplication pour le terme $|D_{EUC}^Q|^2$.

	MR	AR
$64^2 D_{Euc(S_1, S_2)}$ Classique	$[10 \times 64^2]$	$[9 \times 64^2]$
	=	=
	40960	36864
	100%	100%
$64^2 D_{Euc(S_1, S_2)}$ par récursion	$[10 + (1 \times 64^2)]$	$[9 + (1 \times 64^2)]$
	=	=
	4106	4105
	10.02%	11.13%

TABLE 3.2 – Comparaison au niveau complexité calculatoire entre la méthode classique et la méthode par récursion.

4-) Pour décaler le calcul de distance à la colonne suivante, nous effectuons une addition de la valeur du point de départ de la première colonne avec le terme $2 \times h_{j,i}^{(I)}$. Puis, nous effectuons une multiplication pour le terme $|D_{EUC}^I|^2$ afin d'obtenir la distance euclidienne du premier point de la deuxième colonne.

De nouveau, pour déduire les 7 autres points de la colonne, l'étape 3 est réitérée. Notons que pour le calcul d'une distance euclidienne d'un couple $D_{EUC\{S_1, S_2\}}$, les étapes 3 et 4 requièrent 1 AR et 1 MR.

5-) Ce processus est répété jusqu'à détermination des distances euclidiennes de tous les symbols candidats S_1 (resp. S_2) en fixant S_2 (resp. S_1).

Le coût calculatoire pour obtenir l'ensemble des distances euclidiennes $D_{EUC\{S_1, S_2\}}$ pour les $\{64 \times 64\}$ combinaisons possibles des symbols candidats est explicité dans le tableau 3.2.

La première colonne de ce tableau présente le nombre de multiplications réelles nécessaires pour la méthode classique ainsi que pour la méthode par récursion. Dans le reste du document, la méthode classique ayant une complexité calculatoire d'ordre $\{64 \times 64\}$ sera prise comme référence avec une complexité calculatoire de 100% afin d'évaluer nos approches proposées. Le calcul récursif proposé permet une réduction de 90% du nombre de multiplications nécessaires au calcul des distances euclidiennes au sein du détecteur MIMO sans aucune dégradation sur les performances.

3.2 Détecteur simplifié par Calcul Initial d'une Zone Réduite (CIZR) des distances euclidiennes

Dans la section précédente, nous avons proposé une méthode qui nous permet de calculer par récursion toutes les distances euclidiennes des $\{64 \times 64\}$ combinaisons possibles d'une constellation 64-QAM à deux symbols candidats S_1 et S_2 en partant d'un point de la constellation. Grâce à cette méthode, on effectue à chaque fois une simple addition et une multiplication. Cependant, le nombre de multiplications nécessaires pour la méthode par récursion est toujours proportionnel au terme $\{64 \times 64\}$.

Dans cette section, nous allons proposer de réduire l'espace de calcul des distances euclidiennes afin de diminuer le nombre de multiplications réelles. Pour ce faire, nous proposons une méthode de détection basée sur une décomposition de la constellation en des sous-régions statiques ou dynamiques.

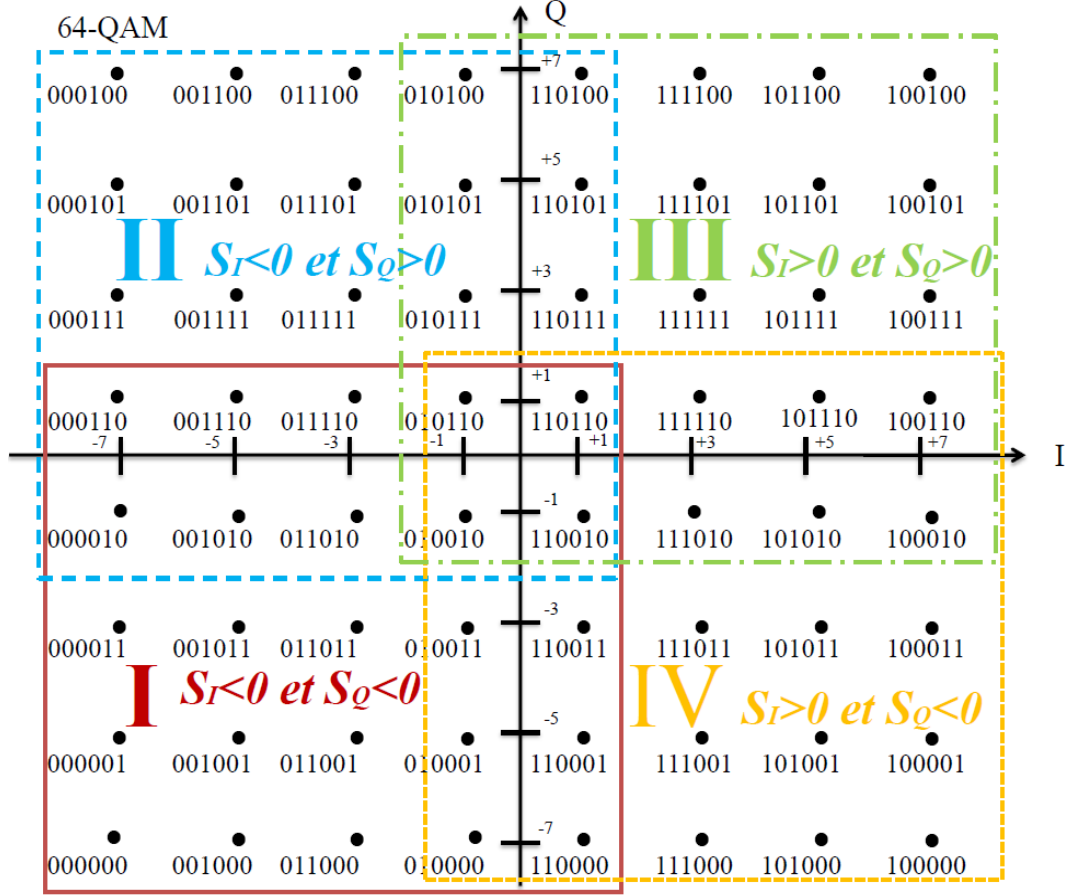


FIGURE 3.2 – Les quatre sous-régions statiques de la constellation 64-QAM

Notons qu'une sous-région statique est délimitée à partir du signe des parties réelle et imaginaire du symbole complexe reçu en appliquant la décision dure comme illustré dans la figure 3.2. Une sous-région dynamique est définie autour du symbole candidat le plus proche du symbole reçu lors d'une décision souple. Pour les méthodes statique et dynamique, nous appliquons la méthode par récursion décrite dans la section (3.1) pour calculer les distances euclidiennes de tous les points des sous-régions sélectionnées.

Dans notre étude, basée sur un multiplexage spatial à $N_t = 2$, nous travaillons sur une détection combinée à partir de deux symboles de la constellation S_1 et S_2 . Dans le reste du document, nous l'appelons "*Détection des distances euclidiennes à deux dimensions*". Le vecteur LRV d'un symbole $S_i \in 64\text{-QAM}$ est défini comme suit :

$$L_{S_i} = [L_{[S_i, \alpha_0]}; L_{[S_i, \alpha_1]}; \dots; L_{[S_i, \alpha_{63}]}]^T \quad (3.7)$$

où $L_{[S_i, \alpha_k]}$ représente la distance euclidienne telle que S_i soit égal au point de la constellation $\alpha_k \in 64\text{-QAM}$ en tenant compte de toutes les valeurs du second symbole candidat dans la constellation, comme illustré dans la figure 3.3. Dans cette figure, nous présentons les distances euclidiennes à deux dimensions des symboles S_1 et S_2 . $\beta_{[i][ii]}$ représente la distance euclidienne telle que S_1 est égal au point de la constellation α_i et S_2 au point de la constellation α_{ii} avec $\{i, ii = 0, 1, 2, \dots, 63\}$ et α_i (resp. α_{ii}) $\in 64\text{-QAM}$.

Notons que pour toutes les méthodes mentionnées dans la suite du document, l'approche

$S_2 \backslash S_1$	$L_{[S_2, \alpha_0]}$	$L_{[S_2, \alpha_1]}$	\dots	\dots	\dots	\dots	$L_{[S_2, \alpha_{63}]}$
$L_{[S_1, \alpha_0]}$	$\beta_{[0][0]}$	$\beta_{[0][1]}$	\dots	\dots	\dots	\vdots	$\beta_{[0][63]}$
$L_{[S_1, \alpha_1]}$	$\beta_{[1][0]}$	$\beta_{[1][1]}$	\dots	\dots	\dots	\vdots	$\beta_{[1][63]}$
$L_{[S_1, \alpha_2]}$	$\beta_{[2][0]}$	$\beta_{[2][1]}$	\dots	\dots	\dots	\vdots	$\beta_{[2][63]}$
\vdots	\vdots	\vdots	\dots	\dots	\dots	\vdots	\vdots
\vdots	\vdots	\vdots	\dots	\dots	\dots	\vdots	\vdots
\vdots	\vdots	\vdots	\dots	\dots	\dots	\vdots	\vdots
$L_{[S_1, \alpha_{63}]}$	$\beta_{[63][0]}$	$\beta_{[63][1]}$	\dots	\dots	\dots	\vdots	$\beta_{[63][63]}$

FIGURE 3.3 – Les distances euclidiennes à deux dimensions pour les symboles S_1 et S_2

consiste à fixer la valeur de S_1 (resp. S_2) à une valeur constante, tout en réduisant la recherche à une sous-région limitée dans l'espace de signal pour S_2 (resp. S_1). Cette étape est répétée jusqu'à ce que toutes les valeurs de S_1 (resp. S_2) soient déterminées. En d'autres termes, la détection du symbole S_1 appliquée sur une sous-région limitée est conditionnée par la valeur du symbole S_2 et vice-versa.

3.2.1 Détecteur simplifié par Région Statique (CIZR-S)

La méthode de détection simplifiée par région statique consiste à décomposer la constellation M QAM en 4 sous-régions [77], comme illustré dans la figure 3.2. Cette décomposition s'effectue selon le signe des parties réelle et imaginaire du symbole complexe reçu. Ainsi, pour une constellation 64-QAM, chaque sous-région se compose de 25 points candidats qui devraient être les symboles les plus proches au symbole reçu [77] (voir la figure 3.4). Cela nous permet de réduire la complexité calculatoire car seul le calcul des distances euclidiennes de la sous-région sera effectué. Ainsi, pour chaque vecteur LRV $L_{[S_1, \alpha_k]}$ de la figure 3.3, nous calculons 25 distances euclidiennes au lieu de 64 par rapport au symbole S_2 et vice-versa. A partir de la figure 3.4, nous construisons le tableau 3.3 qui nous donne le point initial et l'intervalle à considérer dans chaque sous-région.

Par exemple, pour calculer les distances euclidiennes du vecteur $L_{[S_1, \alpha_0]}$ du symbole S_1 illustré dans la figure 3.3, l'application se compose de trois étapes :

1 -) Tout d'abord, on choisit la sous-région particulière au symbole S_2 grâce au signe du symbole reçu en fixant la valeur du symbole S_1 à α_0 . Cette étape nécessite deux comparaisons réelles (CR).

2 -) Ensuite, nous calculons la distance euclidienne du premier point qui se situe dans le coin de la sous-région sélectionnée du symbole S_2 . Ce calcul s'effectue en appliquant l'équation (3.6). Il nécessite

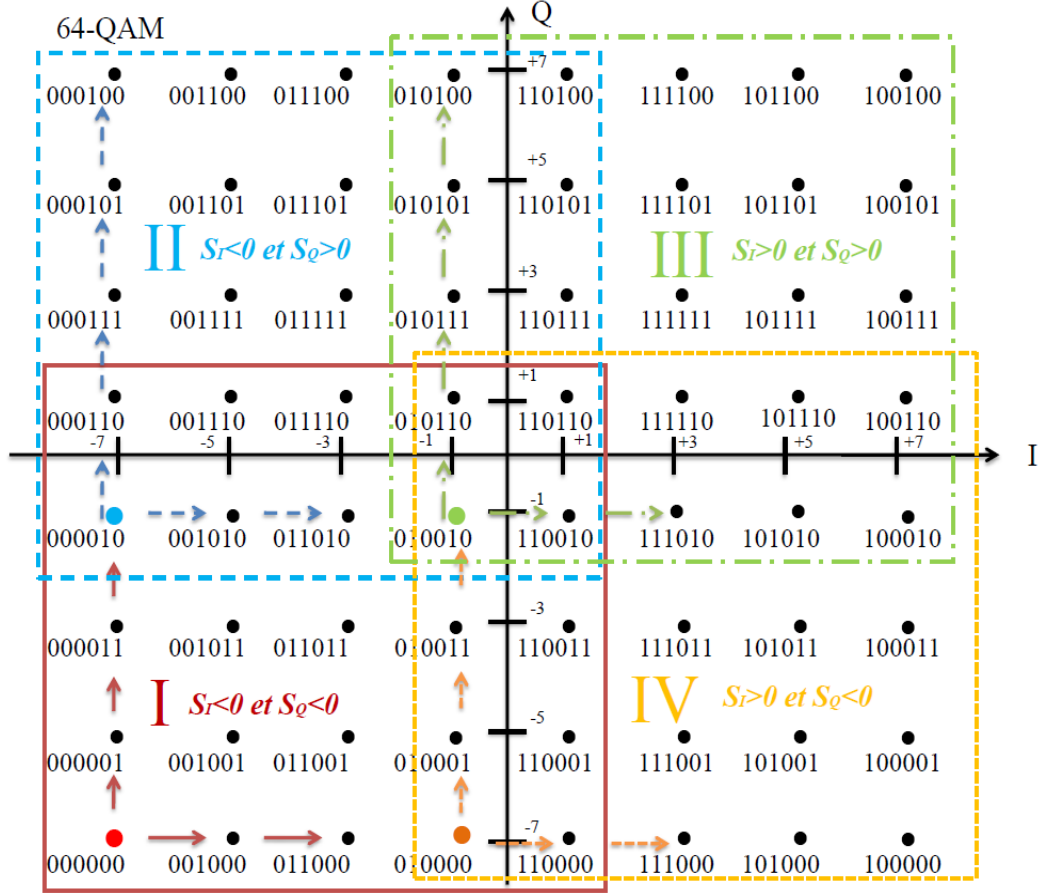


FIGURE 3.4 – Le calcul par récursion des sous-régions statiques de la constellation 64-QAM

Signe du symbol reçu	Point initial	Intervalle de l'axe I	Intervalle de l'axe Q
$S_I < 0; S_Q < 0$	$(-7.0; -7.0)$	$[-7.0 \rightarrow +1.0]$	$[-7.0 \rightarrow +1.0]$
$S_I < 0; S_Q > 0$	$(-7.0; -1.0)$	$[-7.0 \rightarrow +1.0]$	$[-1.0 \rightarrow +7.0]$
$S_I < 0; S_Q < 0$	$(-1.0; -1.0)$	$[-1.0 \rightarrow +7.0]$	$[-1.0 \rightarrow +7.0]$
$S_I < 0; S_Q < 0$	$(-1.0; -7.0)$	$[-1.0 \rightarrow +7.0]$	$[+1.0 \rightarrow -7.0]$

TABLE 3.3 – Le point initial et l'intervalle des points à calculer

10 MR, 9 AR.

3 -) Les 24 distances euclidiennes restantes sont déduites de manière récursive. Chaque calcul de distance nécessite une opération addition et une opération multiplication comme expliqué dans la section précédente. Ce processus est répété pour tous les vecteurs LRV, $L_{[S_1, \alpha_k]}$ tel que $\{S_1 = \alpha^0, \alpha^1, \dots, \alpha^{63}\}$, des symboles S_1 (resp. S_2).

Le coût calculatoire pour obtenir l'ensemble des distances euclidiennes pour les $\{64 \times 25\}$ combinaisons possibles des symbols candidats les plus proches du symbole reçu est explicité dans le tableau 3.4. Il apparaît dans ce tableau une division par deux du nombre de multiplications et d'additions en appliquant la méthode de détection par récursion sur une zone réduite statique $\{64 \times 25\}$ par rapport à la méthode de détection par récursion sur une zone $\{64 \times 64\}$. Au final, en appliquant cette approche, une réduction de 95% du nombre de multiplications nécessaires au calcul des distances

$D_{Euc}(S_1, S_2)$	MR	AR
64^2	$[10 \times 64^2]$	$[9 \times 64^2]$
Classique	= 40960	= 36864
	100%	100%
64^2	$[10 + (1 \times 64^2)]$	$[9 + (1 \times 64^2)]$
par récursion	= 4106	= 4105
	10.02%	11.13%
64×25	$[192 + (64 \times 25)]$	$[338 + (64 \times 25)]$
CIZR-S	=	=
par récursion	1792	1938
	4.37%	5.25%

TABLE 3.4 – Comparaison au niveau complexité calculatoire de la méthode CIZR-S par rapport aux autres méthodes proposées.

euclidiennes est obtenue par comparaison avec la méthode classique $\{64 \times 64\}$.

Les performances de l'algorithme proposé CIZR-S de notre système de réception itératif MIMO 2x2 ayant 32 itérations globales sont données dans la figure 3.5. Le récepteur tire profit de l'algorithme de détection par récursion sur une zone réduite selon la méthode de détection statique. Dans cette figure, nous remarquons que les performances de la méthode statique sont à moins de 0,1 dB de celles du détecteur original tout en réduisant considérablement le nombre d'opérations calculatoires.

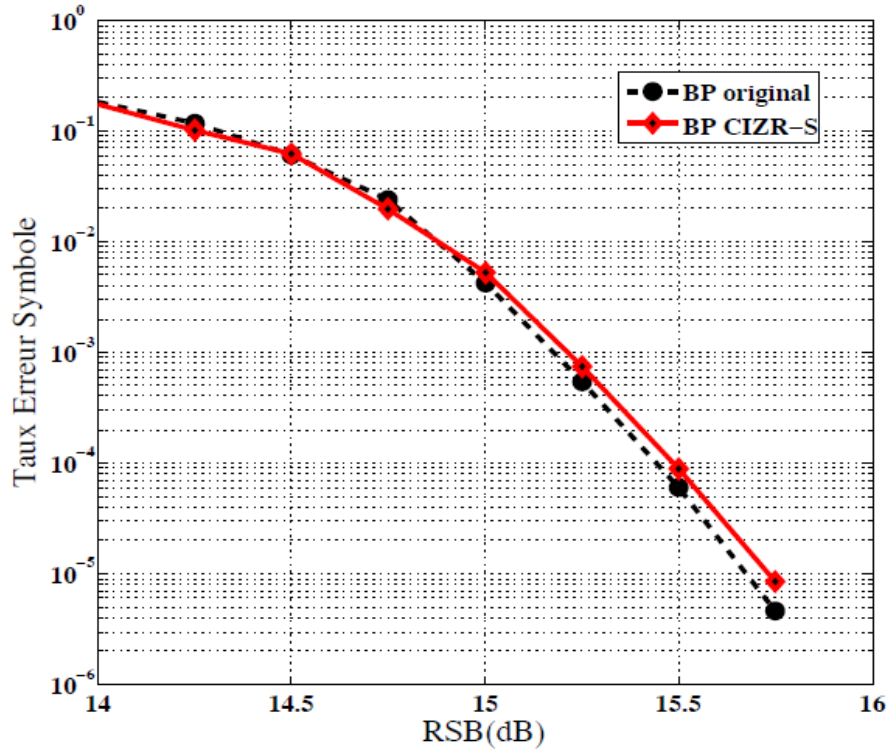


FIGURE 3.5 – Performances de l'algorithme proposé CIZR-S par rapport à l'algorithme de détection originale

Ce travail sur la détection simplifiée par récursion sur une zone réduite statique a fait l'objet d'une publication [78].

3.2.2 Détecteur simplifié par région dynamique (CIZR-D)

Dans cette section, nous proposons une approche de calcul dynamique de la sous-région qui entoure le symbole reçu. En effet, dans le cas où l'on reçoit un symbole à la frontière d'une des 4 sous-régions spécifiées dans la figure 3.4, la méthode par région statique ne calcule pas tous les symboles candidats les plus proches au symbole reçu. Ce cas est illustré dans la figure 3.6, une sous-région dynamique est plus cohérente pour le symbole reçu qu'une sous-région statique.

Pour ce faire, le point le plus proche au symbole reçu dans la constellation 64-QAM doit tout d'abord être estimé. Notre objectif est de trouver une méthode très simple sans avoir besoin de calculer les $\{64 \times 64\}$ combinaisons possibles des symboles candidats et de les comparer entre elles.

Nous rappelons que nous modélisons une chaîne de transmission MIMO composée de $N_t = 2$ antennes à l'émission et de $N_r = 2$ antennes à la réception comme illustré dans la figure 3.7. Les symboles transmis sont de la forme :

$$S_i = \{s = s_I + js_Q | s_I, s_Q \in A_c\} \quad (3.8)$$

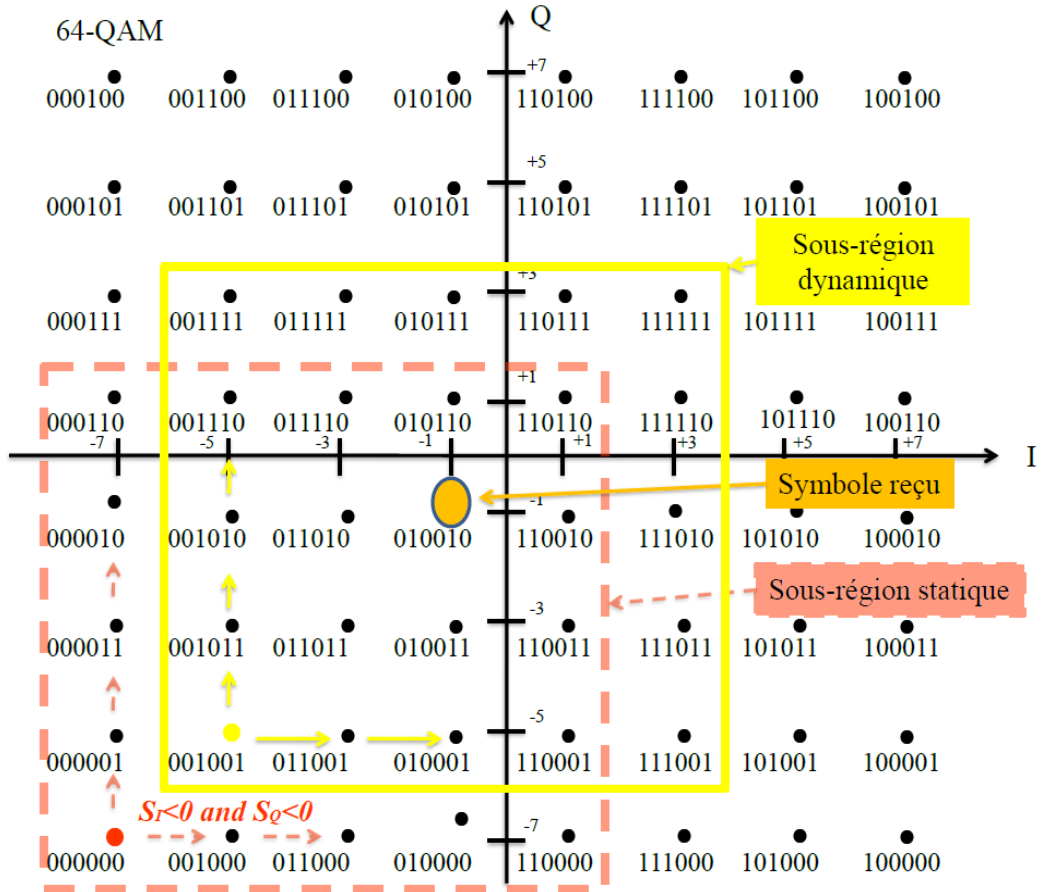


FIGURE 3.6 – Traitement par sous-région dynamique versus statique

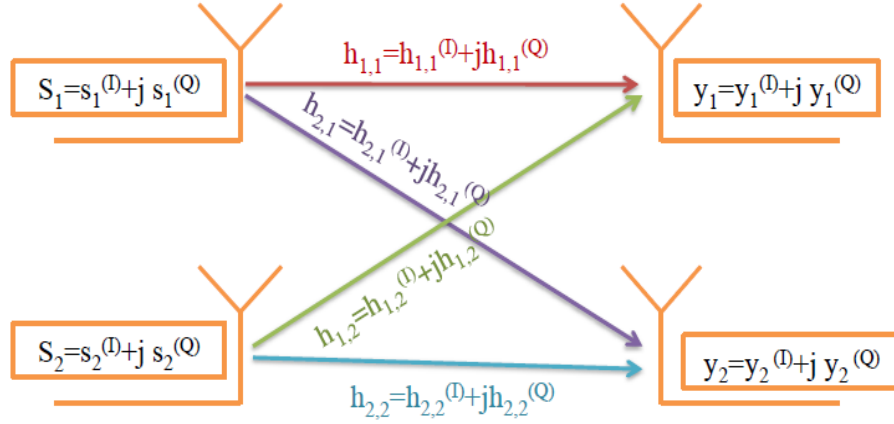


FIGURE 3.7 – Système de communication MIMO 2x2

où A_c est défini comme :

$$A_c = \left\{ \frac{1}{\sigma_s} (-\sqrt{M} + 1 + 2P) \mid P = 0, 1, \dots, \sqrt{M} - 1 \right\} \quad (3.9)$$

et σ_s est le facteur de normalisation et $M = 64$ car la constellation est une 64-QAM.

Nous voulons trouver le symbole candidat S_1 (resp. S_2) le plus proche du symbole reçu par la première antenne en fixant S_2 (resp. S_1) à une valeur d'un symbole candidat 64-QAM comme illustré dans la figure 3.8. Pour ce faire, le symbole reçu à la première antenne est étudié. Une généralisation sera faite dans un second temps.

Le symbole reçu $y_1 = y_1^{(I)} + y_1^{(Q)}$ est exprimé dans l'équation (3.6) sous la forme suivante :

$$\begin{pmatrix} y_1^{(I)} = & (s_1^{(I)} \times h_{1,1}^{(I)}) - (s_1^{(Q)} \times h_{1,1}^{(Q)}) \\ & + (s_2^{(I)} \times h_{1,2}^{(I)}) - (s_2^{(Q)} \times h_{1,2}^{(Q)}) & + n_1^{(I)} \\ y_1^{(Q)} = & (s_1^{(I)} \times h_{1,1}^{(Q)}) + (s_1^{(Q)} \times h_{1,1}^{(I)}) \\ & + (s_2^{(I)} \times h_{1,2}^{(Q)}) + (s_2^{(Q)} \times h_{1,2}^{(I)}) & + n_1^{(Q)} \end{pmatrix} \quad (3.10)$$

En remplaçant $s_1^{(I)}$ et $s_1^{(Q)}$ (resp. $s_2^{(I)}$ et $s_2^{(Q)}$) par leurs expressions en fonction de P^I et $P^Q \in \{0, 1, 2, \dots, \sqrt{M} - 1\}$ (voir l'équation 3.9). L'équation (3.10) peut être réécrite sous la forme suivante :

$$\begin{aligned} y_1^{(I)} &= A + \widehat{h_{1,1}^{(I)}} \cdot P^{I_1} - \widehat{h_{1,1}^{(Q)}} \cdot P^{Q_1} + \widehat{h_{1,2}^{(I)}} \cdot P^{I_2} - \widehat{h_{1,2}^{(Q)}} \cdot P^{Q_2} + n_1^I \\ y_1^{(Q)} &= B + \widehat{h_{1,1}^{(Q)}} \cdot P^{I_1} + \widehat{h_{1,1}^{(I)}} \cdot P^{Q_1} + \widehat{h_{1,2}^{(Q)}} \cdot P^{I_2} + \widehat{h_{1,2}^{(I)}} \cdot P^{Q_2} + n_1^Q \end{aligned} \quad (3.11)$$

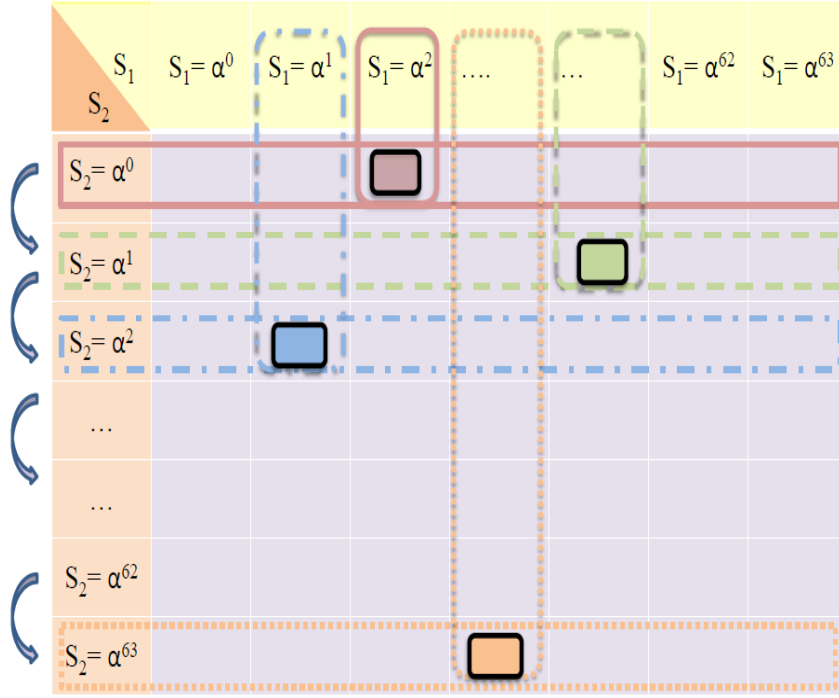


FIGURE 3.8 – Détection du symbole S_1 le plus proche sachant que le symbole $S_2 = \alpha^k$

avec A , B , $\widehat{h_{j,i}^{(I)}}$ et $\widehat{h_{j,i}^{(Q)}}$ représentant des termes constants devant être estimés une seule fois par couple reçu :

$$\begin{aligned} A &= \left[\frac{1}{\sigma_s} \left(-\sqrt{M} + 1 \right) \right] \times \left(h_{1,1}^{(I)} - h_{1,1}^{(Q)} + h_{1,2}^{(I)} - h_{1,2}^{(Q)} \right) \\ B &= \left[\frac{1}{\sigma_s} \left(-\sqrt{M} + 1 \right) \right] \times \left(h_{1,1}^{(Q)} + h_{1,1}^{(I)} + h_{1,2}^{(Q)} + h_{1,2}^{(I)} \right) \end{aligned} \quad (3.12)$$

et

$$\widehat{h_{j,i}^{(I)}} = \left(\frac{2 \cdot h_{j,i}^{(I)}}{\sigma_s} \right), \widehat{h_{j,i}^{(Q)}} = \left(\frac{2 \cdot h_{j,i}^{(Q)}}{\sigma_s} \right) \text{ et } \{P^{I_j}, P^{Q_j} = 0, 1, \dots, 7\}.$$

Par exemple, pour déterminer le symbole candidat S_1 le plus proche sachant que $S_2 = \alpha^k$ comme illustré la figure 3.8, nous appliquons les deux étapes suivantes :

1-) **Détecter P^{I_1} du symbole candidat le plus proche au symbole reçu sachant que $\{S_2 = \alpha^k\}$.** La première étape consiste à calculer le terme P^{I_1} du symbole S_1 pour chaque valeur du symbole S_2 . Pour ce faire, nous résolvons le système d'équations suivantes déduites de l'équation (3.11) :

$$\widehat{h_{1,1}^{(I)}}.P^{I_1} = \overbrace{y_1^{(I)} - A}^{\widehat{y_1^{(I)}}} + \widehat{h_{1,1}^{(Q)}}.P^{Q_1} - \widehat{h_{1,2}^{(I)}}.P^{I_2} + \widehat{h_{1,2}^{(Q)}}.P^{Q_2}(a) \quad (3.13)$$

$$\widehat{h_{1,1}^{(Q)}}.P^{I_1} = \overbrace{y_1^{(Q)} - B}^{\widehat{y_1^{(Q)}}} - \widehat{h_{1,1}^{(I)}}.P^{Q_1} - \widehat{h_{1,2}^{(Q)}}.P^{I_2} - \widehat{h_{1,2}^{(I)}}.P^{Q_2}(b)$$

Pour résoudre ce système d'équations, nous multiplions l'équation (3.13.a), (resp. (3.13.b)), par le coefficient de canal $\widehat{h_{1,1}^{(I)}}$, (resp. $\widehat{h_{1,1}^{(Q)}}$). Puis, les deux équations (a) et (b) sont additionnées :

$$P^{I_1} = \overbrace{\left(\widehat{h_{1,1}^{(I)}}' . \widehat{y_1^{(I)}} + \widehat{h_{1,1}^{(Q)}}' . \widehat{y_1^{(Q)}} \right)}^C - \overbrace{\left(\widehat{h_{1,1}^{(I)}}' . \widehat{h_{1,2}^{(I)}} + \widehat{h_{1,1}^{(Q)}}' . \widehat{h_{1,2}^{(Q)}} \right)}^D . P^{I_2} \quad (3.14)$$

$$+ \overbrace{\left(\widehat{h_{1,1}^{(I)}}' . \widehat{h_{1,2}^{(Q)}} - \widehat{h_{1,1}^{(Q)}}' . \widehat{h_{1,2}^{(I)}} \right)}^E . P^{Q_2}$$

avec $\widehat{h_{1,1}^{(I)}}' = \frac{\widehat{h_{1,1}^{(I)}}}{\widehat{h_{1,1}^{(I)2} + h_{1,1}^{(Q)2}}}$ et $\widehat{h_{1,1}^{(Q)}}' = \frac{\widehat{h_{1,1}^{(Q)}}}{\widehat{h_{1,1}^{(I)2} + h_{1,1}^{(Q)2}}}$ représentant deux termes constants devant être estimés une seule fois pour chaque couple reçu.

Ainsi, une nouvelle expression de P^{I_1} est :

$$P^{I_1} = C - \underbrace{\widehat{D}}_{\hat{D}} . P^{I_2} + \underbrace{\widehat{E}}_{\hat{E}} . P^{Q_2}(d) \quad (3.15)$$

avec C, D et E des constantes. Ces dernières ont une seule expression pour les 64 valeurs possibles du symbole S_2 . Le calcul des termes C, D et E nécessite 6 MR, 5 AR. Par ailleurs, le calcul des termes \hat{D} et \hat{E} nécessite 16 MR car P^{I_2} et $P^{Q_2} \in \{0, 1, 2, \dots, 7\}$ pour couvrir les 64 valeurs possibles du symbole S_2 .

L'étape 1 requiert pour détecter les P^{I_1} des symboles S_1 tel que $\{S_2 = \alpha^0, \alpha^1, \dots, \alpha^{63}\}$: 22 MR, 69 AR.

2-) **Détecter P^{Q_1} du symbole candidat le plus proche au symbole reçu sachant que $\{S_2 = \alpha^k\}$.** La deuxième étape consiste à calculer P^{Q_1} en suivant les mêmes démarches que celles de l'étape 1. Durant cette étape, nous pouvons bénéficier du calcul effectué précédemment. Ainsi, P^{Q_1} a pour expression :

$$P^{Q_1} = \overbrace{\left(\widehat{h_{1,1}^{(I)}}' . \widehat{y_1^{(Q)}} - \widehat{h_{1,1}^{(Q)}}' . \widehat{y_1^{(I)}} \right)}^F + \overbrace{\left(\widehat{h_{1,1}^{(Q)}}' . \widehat{h_{1,2}^{(I)}} - \widehat{h_{1,1}^{(I)}}' . \widehat{h_{1,2}^{(Q)}} \right)}^G . P^{I_2} \quad (3.16)$$

$$- \overbrace{\left(\widehat{h_{1,1}^{(Q)}}' . \widehat{h_{1,2}^{(Q)}} + \widehat{h_{1,1}^{(I)}}' . \widehat{h_{1,2}^{(I)}} \right)}^H . P^{Q_2}$$

La nouvelle expression devient :

$$P^{Q_1} = F + \overbrace{G.P^{I_2}}^{\hat{G}} - \overbrace{H.P^{Q_2}}^{\hat{H}}(e) \quad (3.17)$$

Au niveau complexité calculatoire, cette étape requiert 22 MR, 69 AR.

Finalement, pour détecter les 64 valeurs de P^{I_1} et de P^{Q_1} des symboles S_1 tel que $\{S_2 = \alpha^0, \alpha^1, \dots, \alpha^{63}\}$, nous avons besoin de : 44 MR, 138 AR.

A l'issue de ces calculs, les distances euclidiennes des symboles détectés doivent être calculées.

3-) **Calcul de la distance euclidienne du symbole détecté sachant que $\{S_2 = \alpha^k\}$.** Cette étape consiste à calculer la distance euclidienne de chaque symbole détecté au cours des deux étapes précédentes. Cette distance sera le point initial pour le calcul de toutes les distances euclidiennes des points voisins par la méthode du calcul récursif mentionnée dans la section 3.1. Nous savons que la distance euclidienne a pour expression :

$$D_{EUC(S_1, S_2)} = |D_{EUC}^I|^2 + |D_{EUC}^Q|^2 \quad (3.18)$$

$\{S_1 | S_2 = \alpha^k\}$

En appliquant l'équation 3.11, la distance euclidienne peut être réécrite :

$$D_{EUC(S_1, S_2)} = \overbrace{\left(\overbrace{(y_1^{(I)} - A) - (\widehat{h_{1,1}^{(I)}}.P^{I_1} - \widehat{h_{1,1}^{(Q)}}.P^{Q_1} + \widehat{h_{1,2}^{(I)}}.P^{I_2} - \widehat{h_{1,2}^{(Q)}}.P^{Q_2})}^{D_{EUC}^I} \right)^2}^{D_{EUC}^I{}^2} + \overbrace{\left(\overbrace{(y_1^{(Q)} - B) - (\widehat{h_{1,1}^{(Q)}}.P^{I_1} + \widehat{h_{1,1}^{(I)}}.P^{Q_1} + \widehat{h_{1,2}^{(Q)}}.P^{I_2} + \widehat{h_{1,2}^{(I)}}.P^{Q_2})}^{D_{EUC}^Q} \right)^2}^{D_{EUC}^Q{}^2} \quad (3.19)$$

Les termes $\widehat{y_1^{(I)}}$, $\widehat{y_1^{(Q)}}$ ont été déjà calculés lors de l'étape 1 et 2. Notons que les valeurs possibles des multiplications des coefficients $\widehat{h_{j,i}}$ par P^{I_j} et P^{Q_j} sont limitées. Ainsi, seules 8 valeurs sont possibles pour P^{I_j} et P^{Q_j} . Ce calcul nécessite 32 MR pour couvrir toutes les valeurs possibles de $\{S_2 = \alpha^0, \alpha^1, \dots, \alpha^{63}\}$ comme illustré dans la figure 3.9. Par ailleurs, le calcul de chaque distance euclidienne nécessite 9 AR. À cela, il faut ajouter 2 MR pour le carré des deux termes $D_{EUC}^I{}^2$ et $D_{EUC}^Q{}^2$.

Au final, cette étape requiert $\{(9 \text{ AR} + 2 \text{ MR}) \times 64 + 32 \text{ MR}\}$ pour calculer les distances euclidiennes des 64 symboles détectés pour S_1 sachant que $\{S_2 = \alpha^0, \alpha^1, \dots, \alpha^{63}\}$ (voir la figure 3.9).

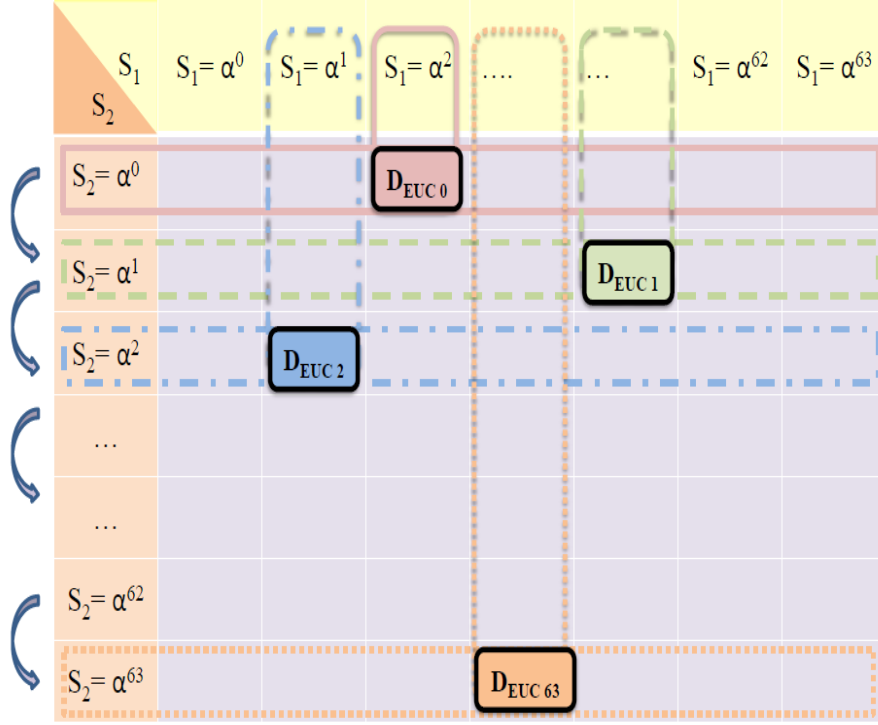


FIGURE 3.9 – Calcul des distances euclidiennes du symbole S_1 sachant que $\{S_2 = \alpha^0, \alpha^1, \dots, \alpha^{63}\}$

Puis, un vecteur des distances euclidiennes allant de D_{EUC0} à D_{EUC63} est construit comme illustré dans la figure 3.9. Ce vecteur représente toutes les distances euclidiennes des symboles S_1 les plus proches du symbole reçu pour chaque valeur fixée $S_2 = \alpha^k$.

4-) **Calcul récursif des distances euclidiennes par exploitation des points voisins de la constellation.** Durant l'étape précédente, nous avons calculé la distance euclidienne du point initial pour chaque symbole détecté S_1 en fixant une valeur pour le symbole candidat S_2 . Dès lors, il est possible de déterminer par un calcul récursif les distances euclidiennes des points voisins des éléments du vecteur $\{D_{EUC0}, D_{EUC1}, \dots, D_{EUC63}\}$. Comme nous l'avons expliqué précédemment, le calcul par récursion une seule distance euclidienne nécessite 1 AR et 1 MR pour le carré de l'un des deux termes D_{EUC}^I et D_{EUC}^Q .

Afin de réduire le nombre de multiplications réelles de notre détecteur, nous proposons de classer les valeurs du vecteur $\{D_{EUC0}, D_{EUC1}, \dots, D_{EUC63}\}$ par ordre croissant sous la forme $\{D_{EUCf_0}, D_{EUCf_2}, \dots, D_{EUCf_{63}}\}$ avec D_{EUCf_0} (resp. $D_{EUCf_{63}}$) représentant la valeur la plus fiable (resp. la moins fiable) parmi les 64 valeurs calculées.

Ainsi, au lieu de calculer récursivement un ensemble de 63 points voisins pour chaque distance euclidienne située dans le vecteur $\{D_{EUC0}, D_{EUC1}, \dots, D_{EUC63}\}$, nous proposons de calculer des ensembles de tailles dynamiques en tenant compte des fiabilités des valeurs D_{EUCf_i} entre elles. C'est à dire, pour les valeurs D_{EUCf_i} les plus fiables, nous calculons récursivement des ensembles ayant des nombres de points voisins plus importants que celles les moins fiables comme illustré la figure 3.10.

Notons que notre objectif est de trouver une combinaison qui nécessite le plus petit nombre de valeurs à calculer avec des performances similaires à celles du détecteur original. Nous avons trouvé

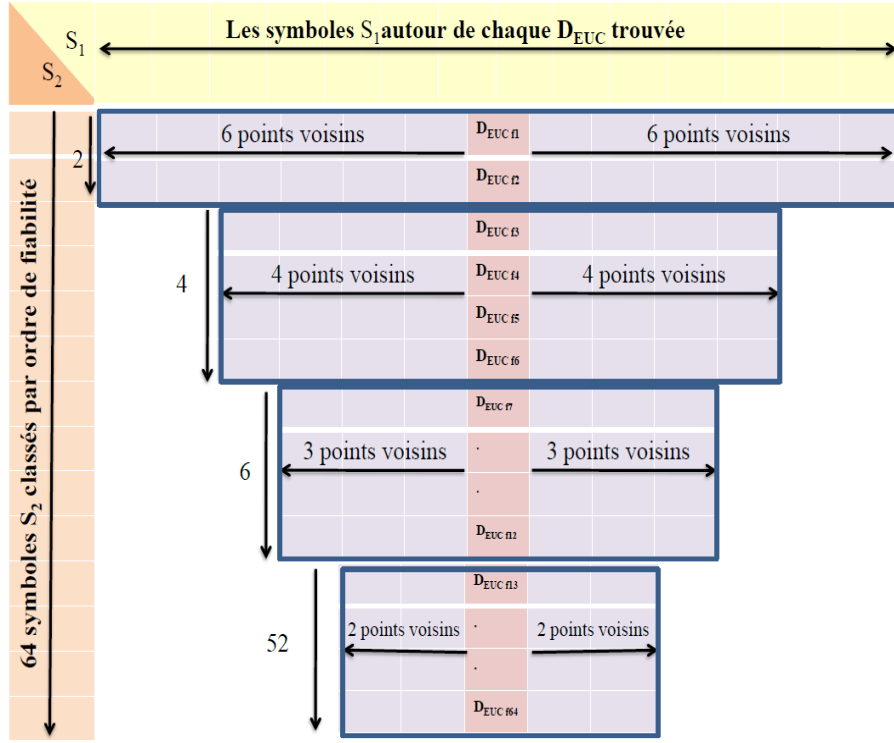


FIGURE 3.10 – Représentation des distances euclidiennes $D_{EUC}f_i$ en tenant compte des fiabilités

par simulation de Monte Carlo une combinaison très pertinente. Elle est illustrée dans la figure 3.10. Elle se décrit de la manière suivante :

- **Pour les deux premières D_{EUC} les plus fiables $\{D_{EUC}f_1, D_{EUC}f_2\}$:** Les 12 points voisins sont obtenus par récursion. Cela nécessite :

$$\{(1MR \times 12) + (1AR \times 12)\} \times 2$$

- **Pour les quatre D_{EUC} qui les suivent, $\{D_{EUC}f_3, \dots, D_{EUC}f_6\}$:** Les 8 points voisins sont obtenus par récursion. Cela nécessite :

$$\{(1MR \times 8) + (1AR \times 8)\} \times 4$$

- **Pour les six D_{EUC} qui les suivent, $\{D_{EUC}f_7, \dots, D_{EUC}f_{12}\}$:** Les 6 points voisins sont obtenus par récursion. Cela nécessite :

$$\{(1MR \times 6) + (1AR \times 6)\} \times 6$$

.

- **Et pour les dernières 52 D_{EUC} en terme de fiabilité, $\{D_{EUC}f_{13}, \dots, D_{EUC}f_{64}\}$:** Les 4 points voisins sont obtenus par récursion. Cela nécessite :

$$\{(1MR \times 4) + (1AR \times 4)\} \times 52$$

.

$D_{Euc}(S_1, S_2)$	MR	AR
64×25	$[192 + (64 \times 25)]$	$[338 + (64 \times 25)]$
	=	=
CIZR-S	1792	1938
par récursion	4.37%	5.25%
	204	714
	+	+
$2 \times 12 + 4 \times 8$	$2 \times 12 + 4 \times 8$	$2 \times 12 + 4 \times 8$
+	+	+
$6 \times 6 + 52 \times 4$	$6 \times 6 + 52 \times 4$	$6 \times 6 + 52 \times 4$
	=	=
CIZR-D	504	1014
par récursion	1.23%	2.75%

TABLE 3.5 – Comparaison au niveau complexité calculatoire de la méthode CIZR-D par rapport à la méthode CIZR-S.

Au total, le coût calculatoire de l'étape 4 est :

$$\{2 \times 12 + 4 \times 8 + 6 \times 6 + 52 \times 4\} \{MR, AR\}$$

Globalement, le coût calculatoire des 4 étapes de la méthode de détection par récursion sur une zone réduite dynamique $\{2 \times 12, 4 \times 8, 6 \times 6, 52 \times 4\}$ est représenté dans le tableau 3.5 en la comparant à la méthode de détection par récursion sur une zone réduite statique $\{64 \times 25\}$. Dans ce tableau, il apparaît une diminution par trois du nombre d'opérations de multiplications et une diminution par deux du nombre d'opérations d'additions en appliquant cette approche par rapport à la méthode de détection par récursion sur une zone statique $\{64 \times 25\}$.

3.2.3 Etude comparative : détecteur de référence *versus* détecteur simplifié CI

Les performances des algorithmes proposés CIZR-S et CIZR-D de notre système de réception itératif MIMO 2x2 ayant 32 itérations globales sont données dans la figure 3.11. Le récepteur tire profit de l'algorithme de détection par récursion sur une zone réduite selon les deux types de détection statique et dynamique.

Dans cette figure, nous remarquons que les performances de la méthode dynamique est plus proche de celles du détecteur original que les performances de la méthode statique tout en réduisant considérablement le nombre d'opérations calculatoires. Cela est dû au fait que la méthode par région statique ne calcule pas tous les symboles candidats les plus proches au symbole reçu. Ainsi, une sous-région dynamique est plus cohérente pour le symbole reçu qu'une sous-région statique.

La comparaison en termes des nombres d'opérations calculatoires entre différentes méthodes de détection est explicitée dans le tableau 3.6. Afin d'évaluer nos approches proposées, nous prenons la méthode classique ayant une complexité calculatoire d'ordre $\{64 \times 64\}$ comme référence avec une complexité calculatoire de 100%.

L'application de la méthode de détection par récursion sur une zone réduite dynamique $\{2 \times 12, 4 \times 8, 6 \times 6, 52 \times 4\}$ permet une division par 8 du nombre d'opérations de multiplications et par 4 celui du nombre d'additions par rapport à la méthode de détection par récursion sur une zone

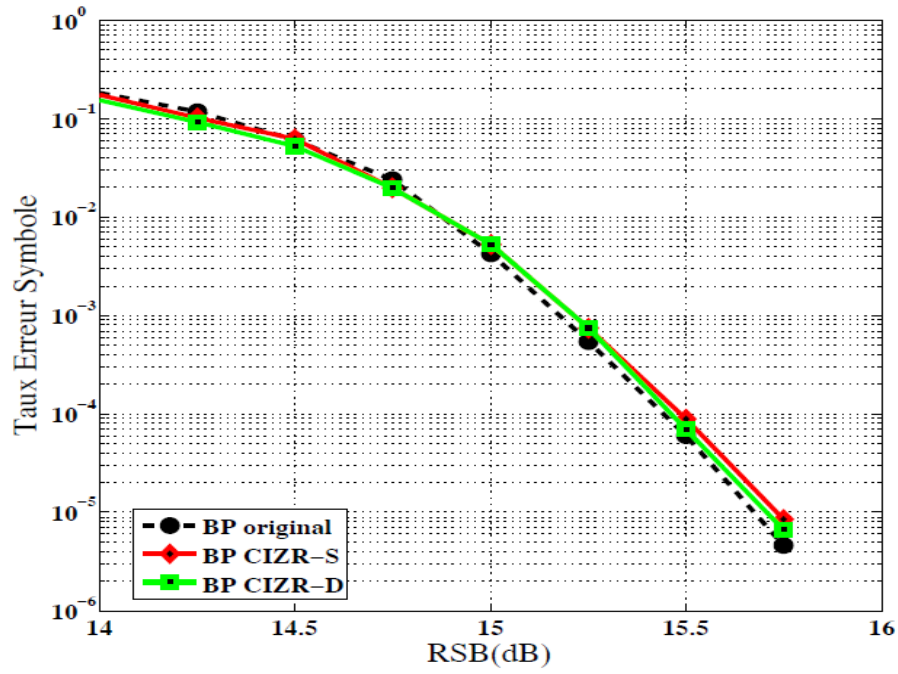


FIGURE 3.11 – Performances de l’algorithme proposé CIZR-D par rapport à l’algorithme de détection CIZR-S

$D_{Euc}(S_1, S_2)$	MR	AR	Nb. de points
64^2 Classique	$[10 \times 64^2]$	$[9 \times 64^2]$	4096
	=	=	
	40960	36864	
64^2 par récursion	$[10 + (1 \times 64^2)]$	$[9 + (1 \times 64^2)]$	4096
	=	=	
	4106	4105	
64×25 CIZR-S par récursion	$[192 + (64 \times 25)]$	$[338 + (64 \times 25)]$	1600
	=	=	
	1792	1938	
$2 \times 12 + 4 \times 8$ + $6 \times 6 + 52 \times 4$ CIZR-D par récursion	$[204 + (2 \times 12 + 4 \times 8) + (6 \times 6 + 52 \times 4)]$	$[714 + (2 \times 12 + 4 \times 8) + (6 \times 6 + 52 \times 4)]$	300
	=	=	
	504	1014	
	1.23%	2.75%	

TABLE 3.6 – Comparaison au niveau complexité calculatoire de la méthode CIZR-S par rapport aux autres méthodes proposées.

$\{64 \times 64\}$.

En appliquant cette approche, une réduction de 98,5% du nombre total de multiplications nécessaires au calcul des distances euclidiennes est obtenue par comparaison avec la méthode classique $\{64 \times 64\}$. Autrement dit, la complexité du calcul global nécessaire est divisée par 80 et par 37 en termes de multiplications et d'additions par rapport à la méthode classique $\{64 \times 64\}$.

Au final, nous remarquons que la méthode de détection par récursion sur une zone réduite dynamique $\{2 \times 12, 4 \times 8, 6 \times 6, 52 \times 4\}$ nécessite une complexité de calcul raisonnable par rapport aux autres méthodes proposées tout en conservant les performances originales de la détection.

3.3 Détecteur simplifié grâce à une Réduction du nombre d'Echanges d'Informations (REI) au cours de chaque inter-itération

Lorsque le nombre de bits par symbole augmente (modulation d'ordre élevé), les mises à jour à l'aide de l'équation (3.2) résultantes d'un processus itératif du détecteur MIMO basé sur l'algorithme BP deviennent très coûteuses au niveau de la complexité calculatoire. En outre, le détecteur MIMO basé sur l'algorithme BP est appliqué à des couples ($N_t = N_r = 2$) de symboles de la constellation.

Dans cette section, nous proposons de réduire la complexité calculatoire du détecteur MIMO basé sur l'algorithme BP par une Réduction de nombre d'Echanges d'Informations (REI) au cours de chaque inter-itération. Pour illustrer notre approche, le processus de mise à jour des distances euclidiennes via un tableau à deux dimensions a été donné dans la figure 3.3. Les lignes (resp. les colonnes) représentent les indices des distances euclidiennes du symbole S_1 (resp. S_2). Ils prennent des valeurs entre 0 et 63 désignées par $L_{S_i} = [L_{[S_i, \alpha_0]}; L_{[S_i, \alpha_1]}; \dots L_{[S_i, \alpha_{63}]}]^T$, dans l'équation (3.1). Les valeurs à l'intérieur du tableau correspondent aux valeurs des distances euclidiennes pour les combinaisons de S_1 et S_2 associées à $\beta[y][x]$.

3.3.1 Structure des messages tronqués

Afin de réduire la complexité calculatoire du traitement itératif, nous proposons de travailler sur les deux types de contribution intervenant lors de la mise à jour des distances euclidiennes comme illustré dans la figure 3.12 : une contribution provenant du décodeur LDPC non-binaire et une contribution provenant des distances euclidiennes initiales.

Contribution provenant du décodeur LDPC non-binaire : L'algorithme de décodage Min-Sum [49] trie sur la base de la fiabilité le vecteur LRV de chaque symbole d'entrée $\{S_i, i = 1, 2, \dots, N_t\}$, composé de 64 valeurs. Notons qu'un vecteur LRV représente une estimation de la probabilité que le symbole S_i soit égal à une valeur α_k comprise dans $\{\alpha_0, \dots, \alpha_{63}\} \in 64\text{-QAM}$. En appliquant l'algorithme Min-Sum [49], le vecteur LRV de chaque symbole peut être tronqué aux n_m valeurs les plus fiables parmi 64 valeurs possibles comme illustré dans la figure 3.13. Nous proposons de profiter de cette approximation afin de diminuer le nombre d'échanges d'informations pour un symbole S_i . Ainsi, un sous ensemble composé de n_m valeurs est transmis au détecteur.

En outre, comme le détecteur fonctionne sur des couples de symboles, il est possible de décoder les couples de nœuds de variable avant de renvoyer les informations vers le détecteur grâce à un ordonnancement de type *shuffle*. Les informations envoyées du décodeur vers le détecteur sont sous forme extrinsèque. Par exemple, lorsque le détecteur met à jour le vecteur LRV du symbole S_1 , l'information extrinsèque provenant du décodeur doit être celle correspondant au vecteur LRV de

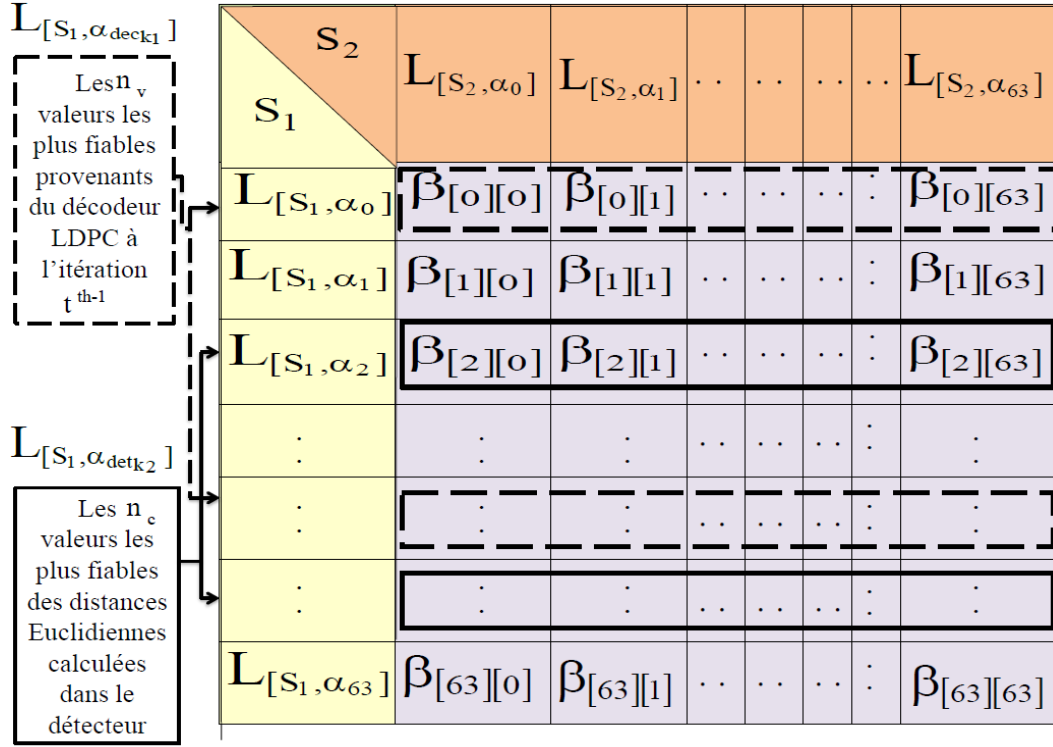


FIGURE 3.12 – Tableau à deux dimensions des distances euclidiennes pour les combinaisons de S_1 et S_2 désignées par $\beta[y][x]$.

symbole S_2 et vice-versa. De plus, nous proposons de tenir compte de la contribution de l'information intrinsèque de chaque symbole.

Nous rappelons que le décodage par l'algorithme Min-Sum est effectué sur des couples de nœuds de variable. Puis, le décodeur fournit un vecteur LRV de n_m valeurs pour chaque symbole décodé classées dans l'ordre croissant en fonction des différentes valeurs de fiabilités. Afin de réduire la complexité calculatoire du processus itératif, il est possible de ne mettre à jour qu'un sous-ensemble composé de n_v vecteurs LRV de S_1 (resp. S_2) par les n_m valeurs des informations extrinsèques de S_2 (resp. S_1) provenant du décodeur. En d'autres termes, l'information intrinsèque de S_1 à la sortie du décodeur LDPC-NB sert à déterminer le sous-ensemble des n_v vecteurs LRV les plus fiables de S_1 qui seront

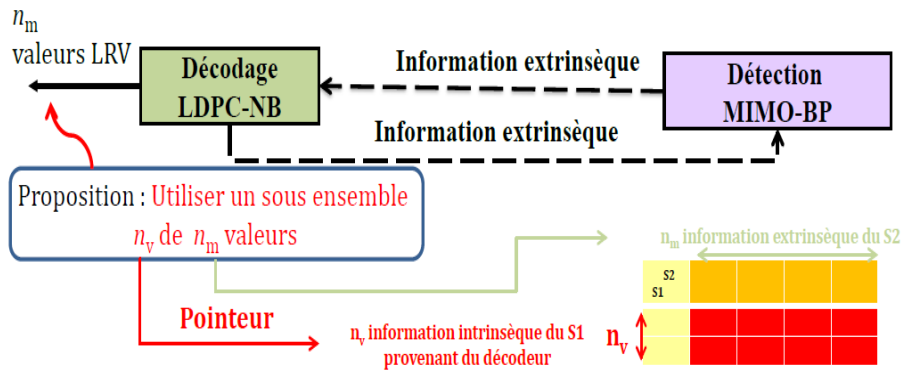


FIGURE 3.13 – Contribution portant sur le décodeur LDPC non-binaire

actualisées par les n_m valeurs extrinsèques du S_2 et réciproquement, comme illustré dans la figure 3.13.

En résumé, le processus de la mise à jour des vecteurs LRV du symbole S_1 est obtenu à partir des informations extrinsèques issues du décodeur pour le symbole S_2 . Afin de maîtriser la complexité calculatoire résultante, seul un sous-ensemble de n_v vecteurs LRV de S_1 est mis à jour. Pour ce faire, le sous-ensemble est déterminé à partir des n_v informations intrinsèques du symbole S_1 provenant du décodeur LDPC-NB. Ce sous-ensemble correspond à la région dec_{n_v} délimitée par les lignes en pointillées dans la figure 3.15. Elle est comprise entre les valeurs $L[S_1, \alpha_{dec_0}]$ et $L[S_1, \alpha_{dec_{n_v-1}}]$ de l'axe des lignes de S_1 et entre les valeurs $L[S_2, \alpha_{dec_{p_0}}]$ et $L[S_2, \alpha_{dec_{p_{n_m-1}}}]$ de l'axe des colonnes de S_2 .

Contribution provenant du calcul des distances euclidiennes initiales : Dans la figure 3.14, nous proposons d'appliquer une approche similaire pour la prise en compte des distances euclidiennes initiales. Pour un symbole particulier S_i , un sous-ensemble de n_c vecteurs LRV des distances euclidiennes est mis à jour par les n_m informations extrinsèques provenant du décodeur. Ce sous-ensemble comprend les n_c vecteurs LRV les plus fiables des distances euclidiennes calculées initialement au sein du détecteur. La région correspondante det_{n_c} est délimitée par la ligne continue sur la figure 3.15. Elle est comprise entre les valeurs $L[S_1, \alpha_{det_0}]$ et $L[S_1, \alpha_{det_{n_c-1}}]$ sur l'axe des lignes de S_1 et entre les valeurs $L[S_2, \alpha_{dec_{p_0}}]$ et $L[S_2, \alpha_{dec_{p_{n_m-1}}}]$ sur l'axe des colonnes de S_2 .

Ainsi, le sous-ensemble de valeurs qui doivent être mises à jour pour le symbole S_1 est noté $\{S_{1upd} = dec_{n_v} \cup det_{n_c}\}$. Dans le cas le moins favorable, la taille de S_{1upd} est égale à $n_{S_{1upd}} = (n_c + n_v)$ lorsque aucun élément commun n'existe entre les régions det_{n_c} et dec_{n_v} . Par conséquent, la taille maximale de l'ensemble des distances euclidiennes à actualiser est $\{n_{S_{1upd}} \times n_m\}$. Cette région est constituée de deux sous-régions en prenant en compte les LRV les plus fiables au niveau des distances euclidiennes du détecteur et au niveau du décodeur. En utilisant cette approche, une nouvelle expression de l'équation (3.2) peut être énoncée pour le symbole $\{S_{iupd} = dec_{n_v} \cup det_{n_c}\}$. Le vecteur LRV a pour expression :

$$L_{[S_i, \alpha_X]} = \log \left(\frac{\sum_{\substack{w=0 \\ S_i=\alpha_X, S_m=\alpha_{dec_{pw}}}}^{n_m-1} \exp(\nu)}{\sum_{\substack{w=0 \\ S_i=\alpha_0, S_m=\alpha_{dec_{pw}}}}^{n_m-1} \exp(\nu)} \right) \quad (3.20)$$

avec

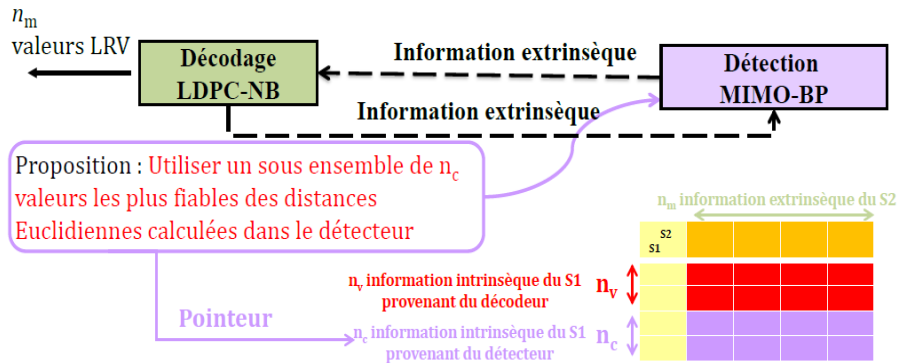


FIGURE 3.14 – Contribution portant sur les calculs initiaux des distances Euclidiennes

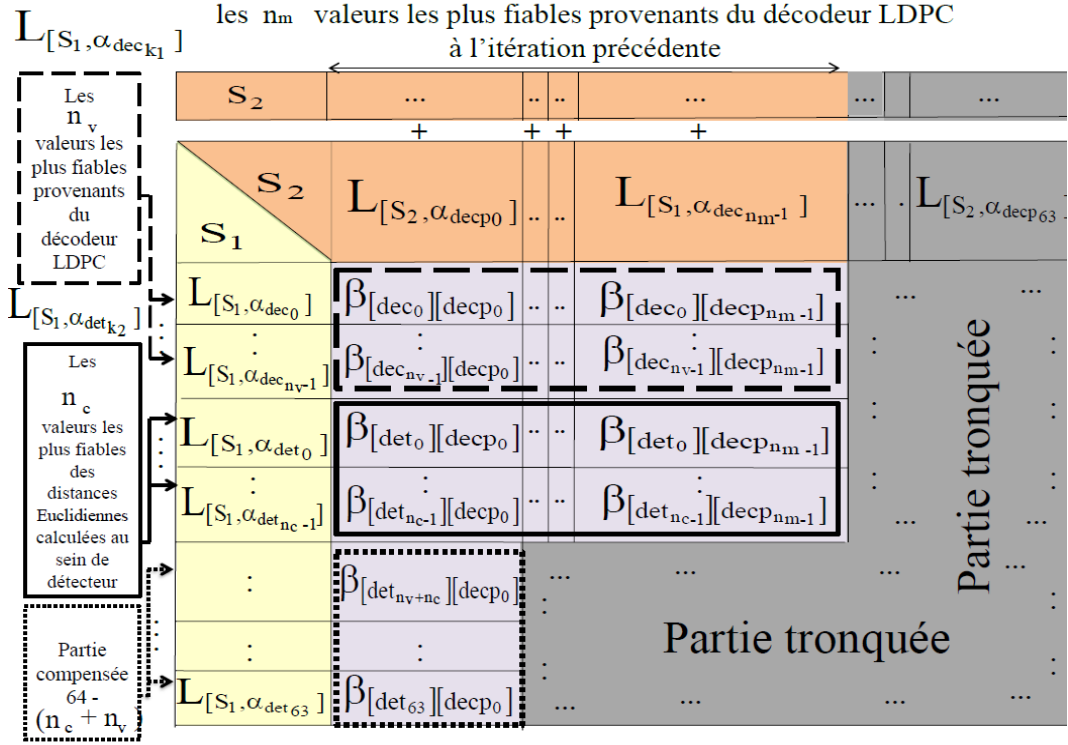


FIGURE 3.15 – Les sous-ensembles des distances euclidiennes à deux dimensions proposés pour la mise à jour.

$$\nu = -\frac{1}{\sigma_w^2} \left| y_j - \sum_i h_{j,i} S_i \right|^2 + \sum_{\substack{(m=1|m \neq i) \\ \alpha_{decw} \in GF(64)}}^{N_t} L[S_m, \alpha_{decw}]$$

α_X correspond à $(\alpha_{det_{k_1}} \cup \alpha_{dec_{k_2}}) \in GF(64)$ avec $(k_1 = 0, \dots, n_c - 1)$, $(k_2 = 0, \dots, n_v - 1)$ et $\alpha_{decw} \in GF(64)$ avec $(w = 0, \dots, n_m - 1)$.

Cette équation peut être également simplifiée comme suit :

$$L[S_i, \alpha_X] \approx \max_{\substack{w=0 \\ S_i = \alpha_X}}^{n_m-1} \left\{ \mu(S_i, S_m) + \sum_{\substack{(m=1|m \neq i) \\ \alpha_{decw} \in GF(64)}}^{N_t} L[S_m, \alpha_{decw}] \right\} - \max_{\substack{w=0 \\ S_i = \alpha_0}}^{n_m-1} \left\{ \mu(S_i, S_m) + \sum_{\substack{(m=1|m \neq i) \\ \alpha_{decw} \in GF(64)}}^{N_t} L[S_m, \alpha_{decw}] \right\} \quad (3.21)$$

Donc, nous réduisons le nombre d'opérations nécessaires lors d'un traitement itératif en limitant la mise à jour des distances euclidiennes au sein du détecteur par les messages *a priori* provenant du décodeur.

3.3.2 Compensation des vecteurs tronqués

La troncature des vecteurs les moins fiables permet une réduction de la complexité calculatoire du traitement itératif. Malheureusement, les performances d'un récepteur MIMO basé sur l'algorithme BP qui applique cette simplification pour le détecteur sont fortement dégradées par comparaison à une détection MV. Cela semble être causé par des cas particuliers où le détecteur et le décodeur ne fournissent pas les mêmes classifications des vecteurs LRV. En outre, l'impact de la contribution du décodeur LDPC semble être plus important lors de la première itération globale. Par exemple, un indice du symbole S_1 ayant des meilleures valeurs de détection n_c peut initier la mise à jour par des informations moins fiables du symbole S_2 provenant du décodeur. A l'inverse, un indice ne faisant pas partie de S_{1upd} peut être mis à jour par des informations extrinsèques plus fiables du symbole S_2 provenant du décodeur. Par conséquent, cette dernière valeur dépassera la précédente en termes de fiabilité lors de la prochaine itération.

Pour résoudre ce problème, nous proposons d'inclure dans la région de traitement une nouvelle zone pour la mise à jour des distances euclidiennes. Cette zone correspond à une valeur de S_2 , c'est à dire une colonne du tableau de la figure 3.15. Le choix de la valeur de S_2 dépend du tri. Il s'agit de la valeur la plus fiable pour le symbole S_2 . La zone correspondante s'ajoute à la région précédemment décrite. Elle est délimitée par les lignes en pointillés courts sur la Fig. 3.13. Elle est comprise entre les valeurs $L[S_1, \alpha_{det_{nc}}]$ et $L[S_1, \alpha_{det_{63}}]$ sur l'axe des lignes de S_1 et une colonne de S_2 .

L'expression de l'équation précédente peut être alors réécrite pour la nouvelle région $(64 - S_{1upd})$:

$$L[S_i, \alpha_Y] \approx \left\{ \mu(S_i, S_m) + \sum_{(m=1|m \neq i)}^{N_t} L[S_m, \alpha_{dec_{p0}}] \right\}_{\substack{S_m = \alpha_{dec_{p0}} \\ S_i = \alpha_Y}} - \left\{ \mu(S_i, S_m) + \sum_{(m=1|m \neq i)}^{N_t} L[S_m, \alpha_{dec_{p0}}] \right\}_{\substack{S_m = \alpha_{dec_{p0}} \\ S_i = \alpha_0}} \quad (3.22)$$

avec α_Y correspondant à $(\overline{\alpha_{det_{k1}} \cup \alpha_{dec_{k2}}})$ et la région des indices de S_1 allant de $(64 - S_{1upd})$.

Dans la sous section suivante, nous allons étudier les diagrammes EXIT au niveau non-binaire qui sont intéressants dans notre contexte pour trouver le bon choix des paramètres n_c , n_v , n_m . Cela nous permet d'évaluer les performances du système simplifié par rapport au système original.

3.3.3 Analyse de la complexité calculatoire et diagrammes EXIT pour l'algorithme de détection proposé

Une comparaison en termes de nombre d'opérations par itération a été faite entre un détecteur MV et le détecteur BP original. Il s'avère que le nombre d'opérations calculatoires pour le détecteur BP croît linéairement en fonction du nombre d'antennes à la réception N_r , tandis que celui du détecteur MV croît exponentiellement en fonction du nombre d'antennes à la réception [69]. Pour illustrer ce constat, les complexités calculatoires requises par les deux détecteurs sont récapitulées dans le tableau 3.7. Par ailleurs, il est possible d'estimer la complexité calculatoire de la version simplifiée du détecteur BP qui a été explicitée dans la section précédente. Ainsi le nombre d'additions pour l'algorithme de détection BP de faible complexité devient :

$$Nb \text{ additions} = \{[(n_c + n_v).n_m^{N_t-1}] + [q - (n_c + n_v)]\}.N_r \quad (3.23)$$

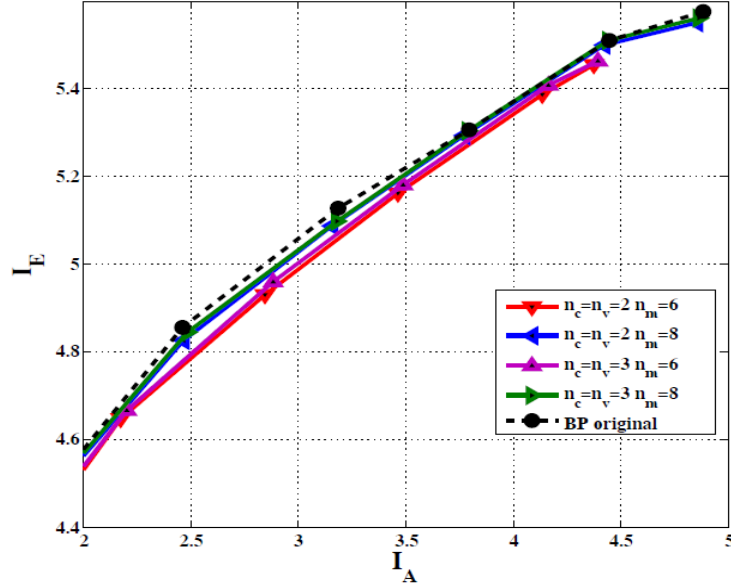


FIGURE 3.16 – Les diagrammes EXIT de l’algorithme BP à faible complexité pour différents profils.

avec $[(n_c + n_v).n_m^{N_t-1}]$ le nombre d’additions pour la mise à jour du sous-ensemble $S_{1_{upd}}$. Le nombre d’additions pour la mise à jour de la colonne supplémentaire $(64 - S_{1_{upd}})$ est $[q - (n_c + n_v)]$.

Le nombre de comparaisons pour le détecteur simplifié est :

$$Nb \text{ comparaisons} = \{[(n_c + n_v).n_m^{N_t-1}] + [q.(n_c + n_v + n_m)]\}.N_r \quad (3.24)$$

où $[(n_c + n_v).n_m^{N_t-1}]$ est le nombre de comparaisons pour la mise à jour du sous-ensemble $S_{1_{upd}}$. Le nombre de comparaisons qui sont nécessaires pour trier les vecteurs les plus fiables de la région det_{n_c} est $[q.n_c]$. Il est à noter que ces valeurs sont uniquement calculées lors de la première itération globale. De même $[q.n_v]$ et $[q.n_m]$ correspondent respectivement aux nombres de comparaisons nécessaires pour les régions dec_{n_v} et $dec_{p_{n_m}}$. En appliquant le principe des messages tronqués, un traitement efficace basé sur la propagation de croyance au niveau de la détection et du décodage est obtenue.

Par ailleurs, le diagramme EXIT [70] est un outil très efficace pour analyser le processus de convergence des systèmes itératifs. Dans [79], la méthode des diagrammes EXIT a été étendue à des processus itératifs non-binaires. Nous avons appliqué cet outil dans notre étude pour analyser la convergence de récepteurs itératifs à faible complexité calculatoire.

La figure 3.16 est également un zoom du diagramme EXIT non-binaire simulé pour des valeurs

Détecteurs	Nb. d’additions	Nb. de comparaisons
MV	$(q.q^{N_t-1}).N_r$	$(q.q^{N_t-1}).N_r$
BP original	$(q.q^{N_t-1}).N_r$	$(q.q^{N_t-1}).N_r$
BP à faible complexité	$[(n_c + n_v).n_m^{N_t-1}]$ + $[(q - (n_c + n_v))].N_r$	$[(n_c + n_v).n_m^{N_t-1}]$ + $[(q.(n_c + n_v + n_m))].N_r$

TABLE 3.7 – Comparaison au niveau des opérations (additions et comparaisons) entre différents détecteurs.

	Nb. d'additions	Nb. de comparaisons	Nb. de points
BP original	8192	8192	4096
BP à faible complexité $\{n_c = n_v = 2, n_m = 6\}$	168	1328	84
BP à faible complexité $\{n_c = n_v = 2, n_m = 8\}$	184	1600	92
BP à faible complexité $\{n_c = n_v = 3, n_m = 6\}$	188	1608	96
BP à faible complexité $\{n_c = n_v = 3, n_m = 8\}$	212	1888	108

TABLE 3.8 – Comparaison au niveau des opérations (additions et comparaisons) entre différents paramètres pour le détecteur BP.

d'informations mutuelles entre 0 et 6. Sur cette figure, I_A représente l'information mutuelle à l'entrée du détecteur qui est la sortie du décodeur. I_E représente l'information mutuelle à la sortie du détecteur correspondant à l'entrée du décodeur. Plusieurs combinaisons de n_c , n_v et n_m ont été étudiées. Notre objectif est de trouver la combinaison qui nécessite le plus petit nombre de valeurs n_c , n_v et n_m avec une performance proche de la version originale du détecteur.

La figure 3.16 montre que l'algorithme BP à faible complexité converge vers les performances de l'algorithme BP original pour $n_c = n_v = 2$ et $n_m = 8$. En outre, cette dernière version nécessite une complexité calculatoire raisonnable par rapport aux autres combinaisons.

Pour illustrer ce constat, la complexité calculatoire globale requise par les différentes combinaisons de l'algorithme BP sont répertoriées dans le tableau 3.8. Pour le profil proposé $n_c = n_v = 2$ et $n_m = 8$, le nombre d'opérations de l'algorithme de détection est divisé respectivement par 40 et par 5 pour les additions et les comparaisons par rapport à l'algorithme BP original.

3.3.4 Etude comparative : détecteur de référence contre détecteur simplifié REI

Les performances de notre système de réception itératif MIMO $\{2 \times 2\}$ ayant 32 itérations globales sont données dans la figure 3.17. Ce système basé sur l'algorithme BP est simplifié par la réduction du nombre d'échanges d'informations (REI) selon différents paramètres. Dans cette figure, nous remarquons que les performances du jeu de paramètre ($n_c = n_v = 2; n_m = 8$) sont très proches de celles du détecteur original tout en réduisant considérablement le nombre d'itérations.

Par conséquent, une division par dix du nombre d'opérations est obtenue en appliquant le jeu $n_c = n_v = 2$ et $n_m = 8$ pour le détecteur BP. La pénalité au niveau des performances est négligeable par rapport aux autres versions de l'algorithme BP. D'autre part, une division par quatre du nombre d'opérations au sein de l'algorithme BP est obtenu par rapport à l'algorithme MV pour des performances similaires avec $NDD = 12$ inter-itérations globales entre le détecteur et le décodeur du système itératif.

De plus, l'algorithme BP à faible complexité peut réduire la latence du récepteur itératif, comme nous le verrons dans le chapitre suivant. La solution algorithmique résultante est une solution particulièrement adaptée pour une implémentation matérielle.

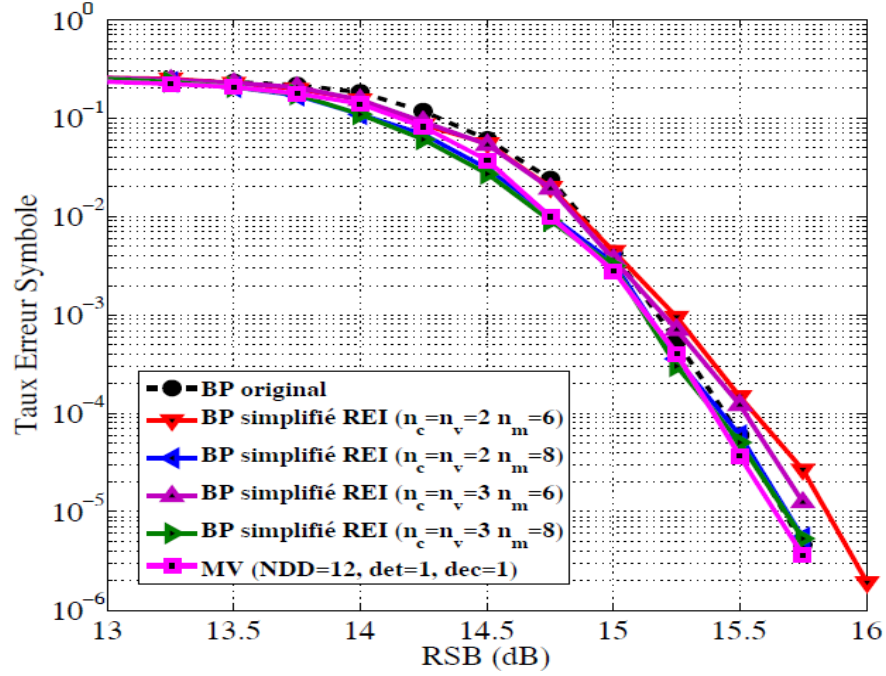


FIGURE 3.17 – Performances du système MIMO itératif pour différents jeux de paramètre pour l'algorithme BP.

3.4 Détecteur simplifié basé sur les techniques CI et REI (CI-REI)

Dans cette section, nous combinons les deux méthodes CI et REI, décrites précédemment, au sein de notre détecteur MIMO itératif. Cette combinaison permet d'aboutir à un algorithme itératif de très faible complexité calculatoire.

Tout d'abord, pour calculer les rapports de vraisemblance (LRV) de tous les points d'une constellation, nous appliquons la méthode de sous-régions dynamiques $\{2 \times 12 + 4 \times 8 + 6 \times 6 + 52 \times 4\}$. Nous exploitons cette approche pour détecter le point initial le plus proche du symbole reçu. Puis, nous calculons tous les points voisins par la méthode de calcul récursif à partir du point détecté. En outre, nous réduisons le nombre d'échanges d'informations entre le détecteur et le décodeur au cours de chaque inter-itération en appliquant la méthode REI avec le jeu de paramètres $n_c = n_v = 2$ et $n_m = 8$ décrit dans la section 3.3.

Les performances du système basé sur une détection de type CI-REI sont présentées dans la figure 3.18. Aucune dégradation significative au niveau des performances n'est observée après l'application de la détection CI-REI. La comparaison en termes de nombres d'opérations calculatoires entre la détection de type CI-REI et la détection originale est explicitée dans le tableau 3.9.

Au niveau de la complexité calculatoire, une réduction d'environ 95% du nombre d'opérations globales par itération est obtenue par rapport au système de référence. De plus, nous constaterons dans le chapitre suivant que notre récepteur MIMO itératif basé sur l'algorithme BP simplifié CI-REI a également une faible latence par rapport à l'algorithme BP original. Cela s'explique par la réduction du nombre d'échanges d'informations durant chaque inter-itération entre le détecteur et le décodeur.

Ce travail sur la détection simplifiée par récursion sur une zone réduite statique a fait l'objet de

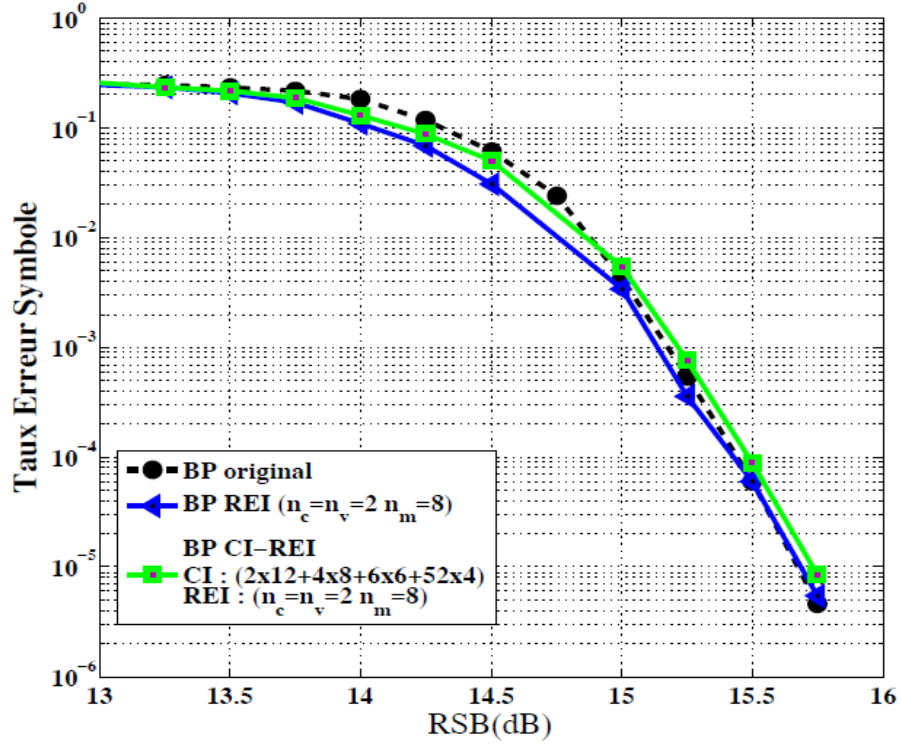


FIGURE 3.18 – Performances de l’algorithme proposé CI-REI par rapport à l’algorithme de détecteur BP original.

Nb. d’opérations par itération	MR	AR	CR
Méthode de détection originale 64^2	40960 100%	36864+8192 =45053 100%	8192 100%
Détection de type CI-REI		1014 + 184 = 1198	1600
CI : $\{2 \times 12 + 4 \times 8 + 6 \times 6 + 52 \times 4\}$	508		
REI : $\{n_c = n_v = 2, n_m = 8\}$	1.23%	2.65%	19.5%

TABLE 3.9 – Comparaison au niveau des opérations de l’algorithme de détection de type CI-REI par rapport à l’algorithme de détection original.

plusieurs publications [78], [80], [81].

3.5 Conclusion

Dans ce chapitre, un récepteur MIMO itératif basé sur l’algorithme de propagation de croyance est étudié en termes de complexité calculatoire. Nous avons étudié la complexité calculatoire du détecteur MIMO à deux niveaux. Le premier niveau est le calcul initial des distances euclidiennes en entrée du détecteur. Le deuxième niveau est celui des échanges d’informations entre le détecteur et le décodeur au cours de chaque inter-itération.

Au premier niveau, nous avons proposé plusieurs méthodes afin de réduire la complexité du calcul des distances euclidiennes pour une constellation d'ordre élevé à deux dimensions dans le cas où $N_t = 2$. Une comparaison des performances et de la complexité calculatoire pour les différentes méthodes proposées a été effectuée. Puis, nous avons proposé une technique pour réduire le nombre d'échanges d'information entre le détecteur et le décodeur dans un contexte itératif. Cette approche réduit considérablement le nombre d'opérations nécessaires (d'environ 95%) au cours de chaque itération sans dégradation au niveau des performances.

En outre, une analyse à l'aide de diagrammes EXIT a été faite dans ce chapitre. Elle a permis de déterminer le jeu de paramètres le plus pertinent au niveau de l'échange d'informations au sein du récepteur itératif. Une analyse de la complexité a montré une réduction d'environ 95% du nombre d'opérations globales par itération au sein du récepteur itératif par rapport à l'algorithme BP de référence. Ainsi, l'algorithme BP à faible complexité peut réduire la latence du récepteur itératif, comme nous le verrons dans le chapitre suivant.

Architecture de réception MIMO itérative basée sur l'algorithme de propaga- tion de croyance

Dans le chapitre précédent, nous avons proposé des solutions algorithmiques afin de réduire la complexité calculatoire des algorithmes associés à un récepteur MIMO itératif basé sur l'algorithme de propagation de croyance tout en limitant les dégradations au niveau des performances. De plus, une étude sur l'organisation des échanges d'information entre le détecteur et le décodeur a été présentée. Le but de cette étude est de définir une architecture globale pour le récepteur itératif permettant un échange efficace d'information par le biais d'un parallélisme favorisant la montée en débit et la maîtrise de la latence du récepteur.

Dans ce quatrième chapitre, nous allons proposer des solutions architecturales pour les différents modules constituant le récepteur itératif. Il est à noter que la définition d'une solution architecturale pour le détecteur MIMO détaille la complexité calculatoire de l'algorithme de détection puis sa mise en oeuvre matérielle. Une reformulation de l'algorithme de détection a été décrite dans le chapitre précédent. Dans ce chapitre, après avoir rapidement présenté un état de l'art des architectures de détection MIMO, nous allons détailler les différents niveaux de l'architecture proposée pour le récepteur itératif. Notons que l'architecture du décodeur LDPC non binaire s'inspire fortement de celle décrite dans [82], [83]. La différence majeure concerne le séquençement. Dans notre contexte, ce dernier est vertical ayant une convergence plus rapide et non horizontal comme dans les travaux de référence.

4.1 Etat de l'art sur l'implémentation d'un détecteur MIMO

Dans la littérature, de nombreuses études se sont intéressées à l'implémentation des algorithmes de détection MIMO. Nous pouvons par exemple citer la détection de type MV [6], [84], la détection de type MMSE [85] et la détection basée sur la propagation de croyance [7].

Il est communément admis qu'une détection de type MV classique est optimale en termes de performance. Cependant, sa complexité calculatoire est incompatible avec une implémentation matérielle. En effet, cette dernière augmente exponentiellement selon l'ordre de la modulation. C'est la raison pour laquelle les détecteurs de type MV sont généralement basés sur des approximations comme dans [86], [11]. Une implémentation de l'algorithme de détection MV classique a été présentée par Burg dans [87]. Par ailleurs, il existe des détecteurs MIMO ayant une complexité polynomiale, à savoir le détecteur basé sur le décodage sphérique dans [88],[89]. Dans [90], Koike a proposé une détection par sphère en remplaçant la métrique euclidienne par d'autres métriques afin de réduire le nombre total de multiplications.

Plusieurs études [91], [92] ont montré que les algorithmes de type MMSE permettent de réduire la complexité calculatoire et d'atteindre des performances s'approchant de celles d'un détecteur de type MV. Cependant, les détecteurs MIMO de type MMSE reposent sur une inversion matricielle. C'est pourquoi, la majorité des travaux consacrés à ce type de détection [93], [94], se sont intéressés aux techniques d'inversion matricielle pour faciliter l'implémentation.

Récemment, d'autres solutions architecturales comme dans [68], [95], ont été proposées afin de maîtriser la complexité calculatoire des détecteurs MIMO. Parmi ces dernières, la technique de détection MIMO basée sur la propagation de croyance semble être une approche offrant des atouts indéniables pour une implémentation matérielle. Le détecteur MIMO basé sur l'algorithme de propagation de croyance offre un avantage certain au niveau de la complexité calculatoire par rapport à un détecteur basé sur le maximum de vraisemblance. L'algorithme BP a été appliqué avec succès pour une détection MIMO ayant une complexité calculatoire linéaire dans [68]. Il a été montré dans [69] que ce type d'algorithme permet d'avoir une détection avec une complexité linéaire en fonction du nombre d'antennes contrairement à une détection basée sur l'algorithme MV qui croît exponentiellement avec le nombre d'antennes.

En comparant les différents types de détection, nous constatons que l'utilisation d'une modulation de faible ordre comme la modulation QPSK a permis l'implémentation de la détection MV classique. Le détecteur de type MMSE permet, quant à lui, l'utilisation d'une modulation d'ordre plus élevé, notamment la modulation 16-QAM, tout en gardant une complexité comparable à celle d'un détecteur MV classique. Le détecteur de type BP peut convenir, quant à lieu, à des modulations d'ordre élevé comme la modulation 64-QAM.

Par rapport aux autres types de détecteurs, l'algorithme de détection de type BP semble a priori plus facile à implémenter sur des circuits matériels de type FPGA. En effet, il apparaît que le détecteur de type BP offre un meilleur compromis complexité/performance [11], [69]. Le détecteur BP est donc un bon candidat pour concevoir un système MIMO itératif de faible complexité.

Notre étude s'est focalisée sur l'implémentation de l'algorithme de détecteur MIMO basé sur la propagation de croyance ayant une complexité calculatoire réduite. Elle a cependant concerné l'ensemble du récepteur de type BP.

4.2 Architecture du récepteur MIMO itératif

Cette section décrit l'architecture du récepteur MIMO itératif. Nous présentons brièvement les différents éléments constituant le récepteur, ces derniers seront détaillés dans les sections suivantes. Notons que dans la suite de notre étude, un cycle d'exécution correspond au temps nécessaire pour traverser deux multiplieurs et un additionneur de deux nombres réels. Ce choix a pour but de concevoir une architecture efficace au niveau des ressources. Il sera justifié lors de la description du détecteur.

La figure 4.1 donne une vue globale de l'architecture. Ce niveau se compose de deux grandes parties disjointes. La partie supérieure est la représentation schématique de l'architecture du détecteur MIMO décrit dans la section 3.4. Tous les blocs principaux qui permettent de traiter les symboles reçus et les symboles candidats sont donnés. La partie inférieure correspond à l'architecture de décodage d'un code LDPC-NB indiquant le processeur pour les nœuds de variable et le processeur pour les nœuds de parité. Cette étude exploite l'architecture de décodeur LDPC non binaire présentée dans [82], [83]

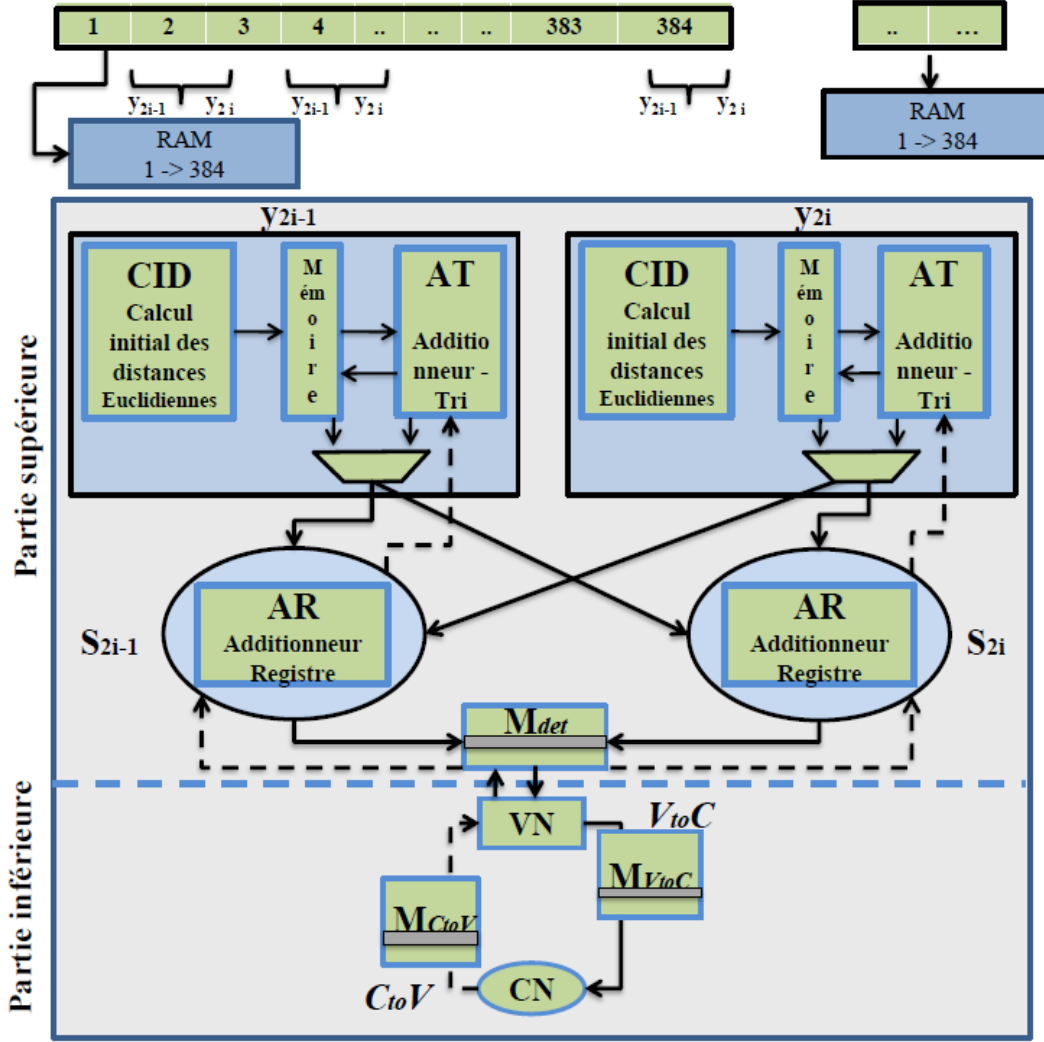


FIGURE 4.1 – Architecture globale du récepteur MIMO

dont le séquençement est modifié pour une adéquation judicieuse avec le détecteur.

Le détecteur reçoit les symboles transmis par le canal. Dans les chapitres précédents, nous avons étudié des trames composées de 384 symboles. Les symboles reçus sont représentés sous formes des couples, y_{2i-1} et y_{2i} avec $i = 1, 2, \dots, 192$, au sein du récepteur grâce au multiplexage spatial. Puisque notre récepteur est itératif, nous avons besoin de mémoires RAM pour stocker les symboles qui sont reçus au cours du traitement des symboles de la trame courante.

L'architecture globale de notre récepteur MIMO considère un couple de nœuds de symboles reçus y_{2i-1} et y_{2i} associé au couple des symboles candidats S_{2i-1} et S_{2i} , grâce à un multiplexage spatial avec $N_t = 2$ antennes d'émission et $N_r = 2$ antennes de réception. Le rôle des nœuds de symboles reçus y_{2i-1} et y_{2i} est de calculer les distances euclidiennes initiales sous forme de vecteurs LRV de tous les symboles candidats. Ce calcul se fait via le bloc CID. Puis, les vecteurs LRV calculés sont sauvegardés dans une mémoire afin de mettre à jour les informations durant chaque inter-itération entre le détecteur et le décodeur. Au cours de la première inter-itération, les informations sauvegardées sont envoyées directement aux nœuds de symboles candidats S_{2i-1} et S_{2i} sans passer par le bloc AT (Additionneur-Tri). Le bloc AT a pour fonction de mettre à jour les in-

formations sauvegardées au sein du détecteur par les informations extrinsèques provenant du décodeur.

Les calculs au sein des nœuds associés aux symboles candidats sont effectués à l'aide d'un bloc AR (Additionneur-Registre). Ils permettent de tenir compte des informations *a priori* des autres symboles reçus. Ensuite, les messages mis à jour sont sauvegardés dans une mémoire M_{det} afin de permettre leur traitement par le décodeur LDPC non binaire.

Au sein du décodeur, nous allons utiliser un seul processeur pour les nœuds de variable connecté à un seul processeur pour les nœuds de parité. La première étape au sein du décodeur LDPC non binaire est de mettre à jour le processeur des nœuds de variable en profitant de l'information extrinsèque provenant du détecteur MIMO. Cette dernière est sauvegardée dans la mémoire M_{det} . Les informations issues du processeur du nœud de parité à l'itération précédente sont également exploitées. Elles sont sauvegardées dans la mémoire M_{CtoV} . Puis, l'information *a posteriori* du processeur des nœuds de variable sera sauvegardée dans $dv - 1$ mémoires M_{VtoC} . Elle est ensuite exploitée par le processeur associé aux nœuds de parité.

Le processeur des nœuds de parité CN connecté au processeur des nœuds de variable VN traite les informations reçues selon la matrice de parité dont les éléments non-nuls sont définis sur le corps de Galois. Puis, les informations extrinsèques seront envoyées aux $dc - 1$ mémoires M_{CtoV} connectés à CN. Après le processus de décodage, les informations extrinsèques sont renvoyées vers le détecteur. Les messages provenant du décodeur sont exploités comme des informations *a priori* pour une

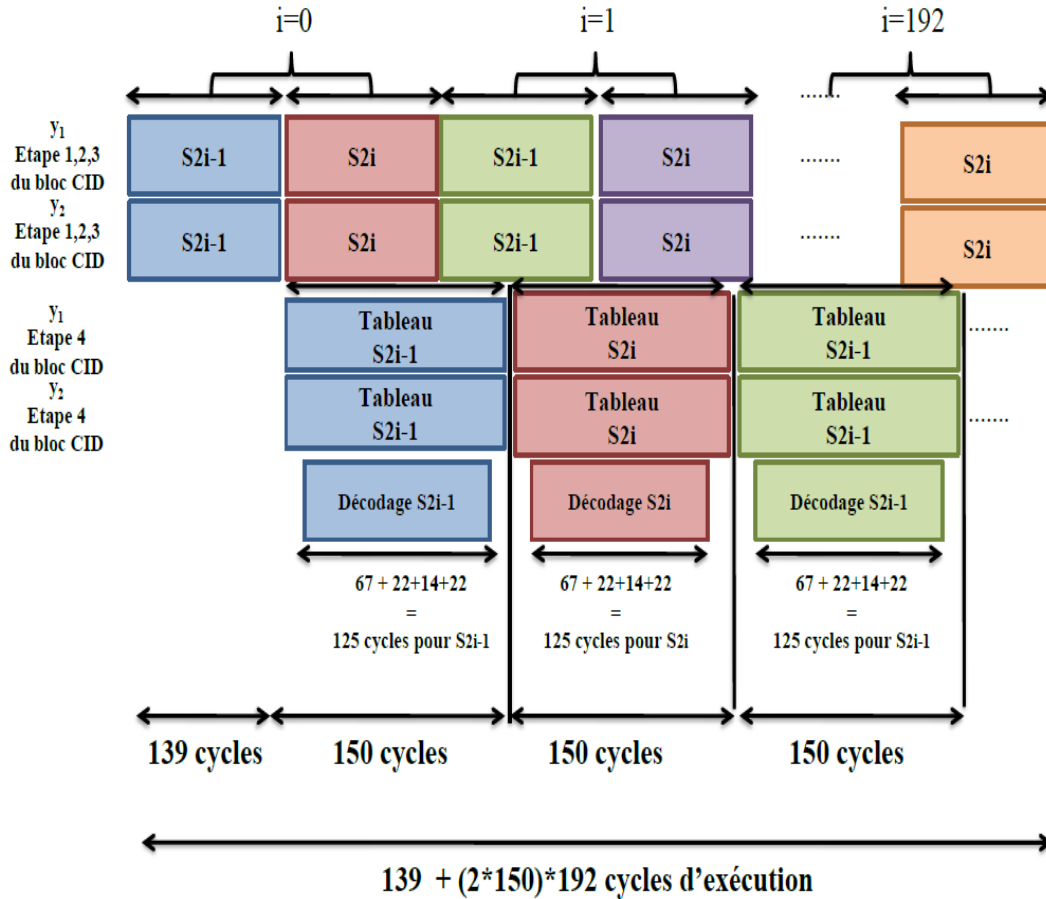


FIGURE 4.2 – Découpe temporelle de la première inter-itération entre le détecteur et le décodeur

nouvelle détection.

Dans la suite de cette section, nous allons donner une vue globale reposant sur le nombre de cycles d'exécution nécessaires lors de l'exécution de la première inter-itération et des autres inter-itérations du récepteur MIMO. Il est à noter que le calcul de cycles d'exécution sera détaillé dans les sections suivantes.

Une découpe temporelle de la première inter-itération du récepteur MIMO est détaillée dans la figure 4.2. Cette découpe montre l'ordonnancement des différentes opérations réalisées par l'ensemble des blocs du détecteur MIMO ainsi que le décodage de chaque symbole candidat. Les sous-blocs du bloc CID qui calculent les distances euclidiennes de tous les symboles candidats nécessitent 139 cycles d'exécution à l'issue desquels les vecteurs LRV envoyés au décodeur sont disponibles.

Après ces 139 cycles d'exécution, le décodeur commence à décoder le premier symbole candidat qui nécessite 125 cycles d'exécution. En parallèle, le bloc 4 s'exécute pour calculer par récursion les points voisins des distances euclidiennes. cela qui nécessite 150 cycles d'exécution. De manière similaire, nous traitons tous les symboles de la trame courante en détectant et décodant de manière conjointe.

Par conséquent, nous avons besoin d'une latence initiale de 139 cycles d'exécution au sein du détecteur à l'issue de laquelle les LRV envoyés au décodeur sont disponibles. Puis, 150 cycles d'exécution sont nécessaires pour détecter et décoder conjointement les informations. Ainsi, la durée de la première inter-itération est de $\{139 + (150 \times 2(\text{couple})) \times 192(\text{symboles})\}$ cycles d'exécution.

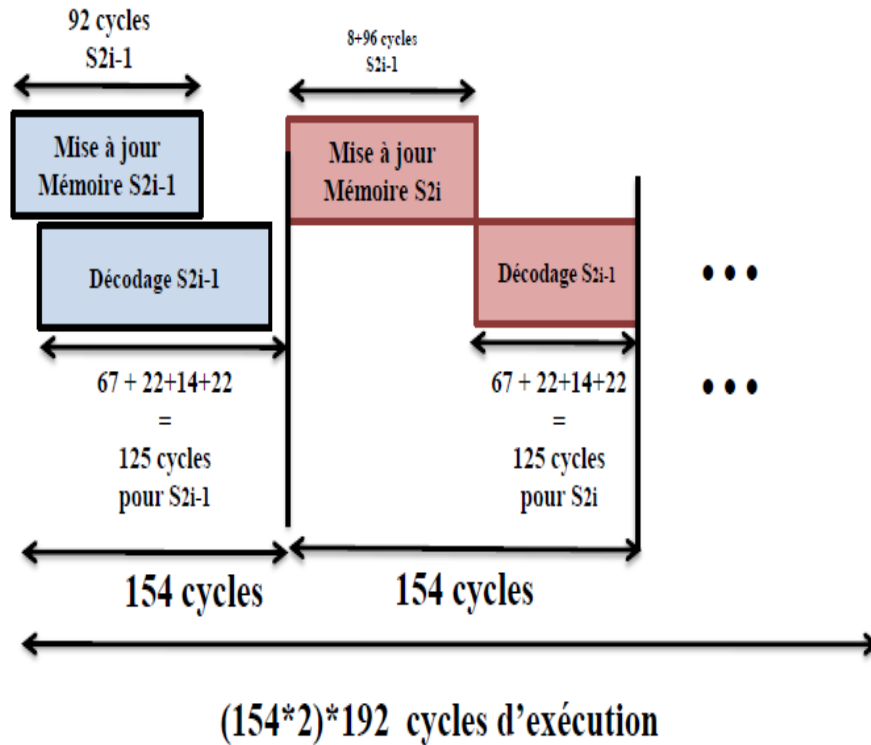


FIGURE 4.3 – Découpe temporelle des autres inter-itérations, distinctes de la première, entre le détecteur et le décodeur

La découpe temporelle des autres inter-itérations, distinctes de la première, du récepteur MIMO est décrite dans la figure 4.3. Notons qu’au cours de la première inter-itérations du récepteur MIMO, le calcul initial des distances euclidiennes est sauvegardé dans la mémoire. Donc, au cours des autres inter-itérations, nous n’avons plus besoin de passer par le bloc CID. Nous mettons uniquement à jour l’informations sauvegardée dans le bloc Mémoire. Ce traitement nécessite 92 cycles d’exécution à l’issue desquels les 64 valeurs LRV devant être envoyées au décodeur sont disponibles.

En revanche, nous n’avons plus besoin d’attendre la génération des 64 valeurs du détecteur MIMO. Dès que nous recevons la première valeur LRV générée par le détecteur, le décodeur commence à décoder le symbole candidat. Ensuite, 150 cycles d’exécution sont donc nécessaires pour détecter et décoder conjointement les symboles. L’étape de mise à jour de chaque symbole nécessite 154 cycles d’exécution. La durée totale pour la mise à jour des autres inter-itérations, distinctes de la première, est de $\{(154 \times 2(\text{couple})) \times 192(\text{symboles})\}$ cycles d’exécution.

4.3 Architecture du détecteur MIMO souple

Dans notre étude architecturale, le détecteur MIMO souple retenu repose sur le principe de détection décrit dans la section 3.4. Il correspond à une détection simplifiée basée sur les techniques CI et REI (CI-REI). Nous rappelons que la détection souple consiste à retrouver des informations souples à partir des symboles transmis par l’émetteur. Cela correspond au calcul du rapport de vraisemblance (LRV) du point initial d’une constellation en considérant la méthode de sous-régions dynamiques $\{2 \times 12 + 4 \times 8 + 6 \times 6 + 52 \times 4\}$. Ainsi, tous les points voisins du point détecté sont obtenus par un calcul récursif.

Dès lors, le calcul initial des distances euclidiennes par la méthode dynamique constitue le bloc critique autour duquel le détecteur est construit. C’est pourquoi il a nécessité une étude approfondie dans le chapitre précédent. Ce travail a abouti à la proposition d’un calcul ayant une complexité raisonnable. Nous rappelons également qu’afin de simplifier l’architecture du récepteur MIMO-BP itératif, une reformulation permettant de réduire le nombre d’échanges d’informations au cours de chaque inter-itération a été proposée dans la section 3.3.

A partir des descriptions précédentes de l’algorithme de détection, nous allons tout d’abord décomposer l’architecture de notre détecteur sous trois niveaux dépendant de leurs fonctionnements.

L’architecture globale du détecteur MIMO est détaillée dans la figure 4.4. La première fonction effectuée par cette architecture est de calculer les distances euclidiennes sur deux dimensions de tous les symboles candidats pour chaque symbole reçu. Cette fonction est faite par le bloc CID qui représente l’élément critique au niveau de la complexité matérielle du détecteur. Les distances calculées sont sauvegardées dans une mémoire RAM (*Random Access Memory*) afin de les mettre à jour à chaque inter-itération.

Ensuite, un bloc AT (Additionneur-Tri) est nécessaire pour additionner les informations extrinsèques provenant du décodeur à celles qui sont associées dans la mémoire. Puis, les sommes sont triées selon leur fiabilité. Il est à noter qu’au cours de la première inter-itération entre le détecteur et le décodeur, nous n’avons pas des informations extrinsèques provenant du décodeur. Donc, les informations intrinsèques sauvegardées dans la mémoire seront envoyées directement aux nœuds de symboles candidats grâce à un multiplexeur ajouté à l’architecture.

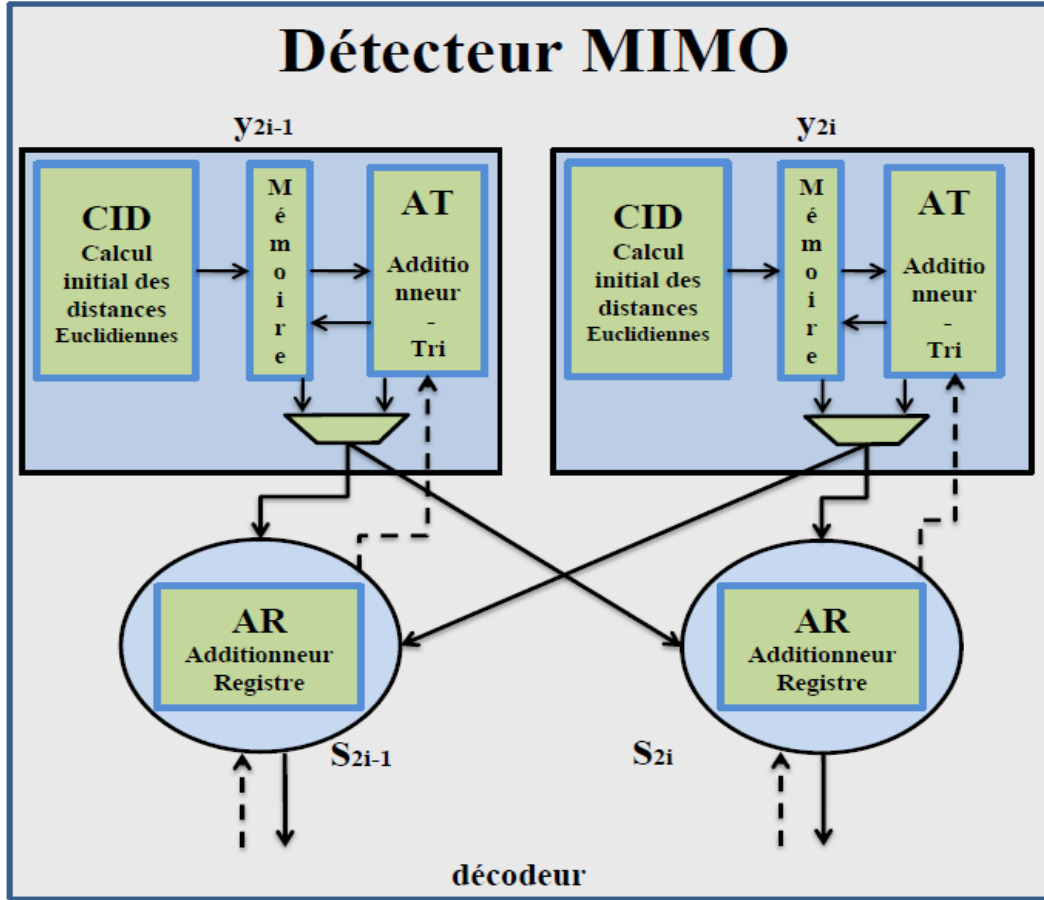


FIGURE 4.4 – Architecture globale du récepteur MIMO itératif

Les calculs au sein des nœuds associés aux symboles candidats sont effectués au sein d'un bloc AR (Additionneur-Registre). Il se compose d'un additionneur et d'un registre de 64 valeurs. Ils permettent de tenir compte des informations *a priori* des autres symboles reçus. Ainsi, les informations associées aux symboles candidats sont mises à jour à partir des informations provenant des autres symboles reçus et des informations extrinsèques provenant du décodeur lors de l'itération précédente. L'informations souple associée aux symboles sont ensuite envoyées au décodeur.

4.3.1 Bloc CID (Calcul Initial des Distances euclidiennes)

Ce bloc a été décomposé en plusieurs sous-blocs. Il est utilisé uniquement au cours de la première inter-itération entre le détecteur et le décodeur. Son rôle est de calculer les distances euclidiennes initiales des 64 vecteurs LRV à deux dimensions entre le symbole reçu et tous les symboles candidats d'une constellation. En conséquence, ce bloc est l'élément critique au niveau de la complexité matérielle autour duquel le détecteur est construit. Ainsi, pour calculer les distances euclidiennes initiales, la méthode de sous-régions dynamiques $\{2 \times 12 + 4 \times 8 + 6 \times 6 + 52 \times 4\}$ est appliquée. Dans cette section, nous présentons les différents éléments constituant le bloc CID.

La figure 4.5 présente l'architecture du bloc CID. Cette dernière a été décomposée en quatre sous-blocs. Le parallélisme a été favorisé dans les blocs 1, 2 et 4 pour atteindre un débit important en sortie du détecteur. Afin de déterminer simultanément les coordonnées (I, Q) des distances euclidiennes, nous proposons de calculer la partie réelle I via les blocs 1, 2, 3 et 4. En parallèle, nous calculons la partie imaginaire Q via des blocs similaires 1', 2', 3 et 4'. Notons que nous n'avons

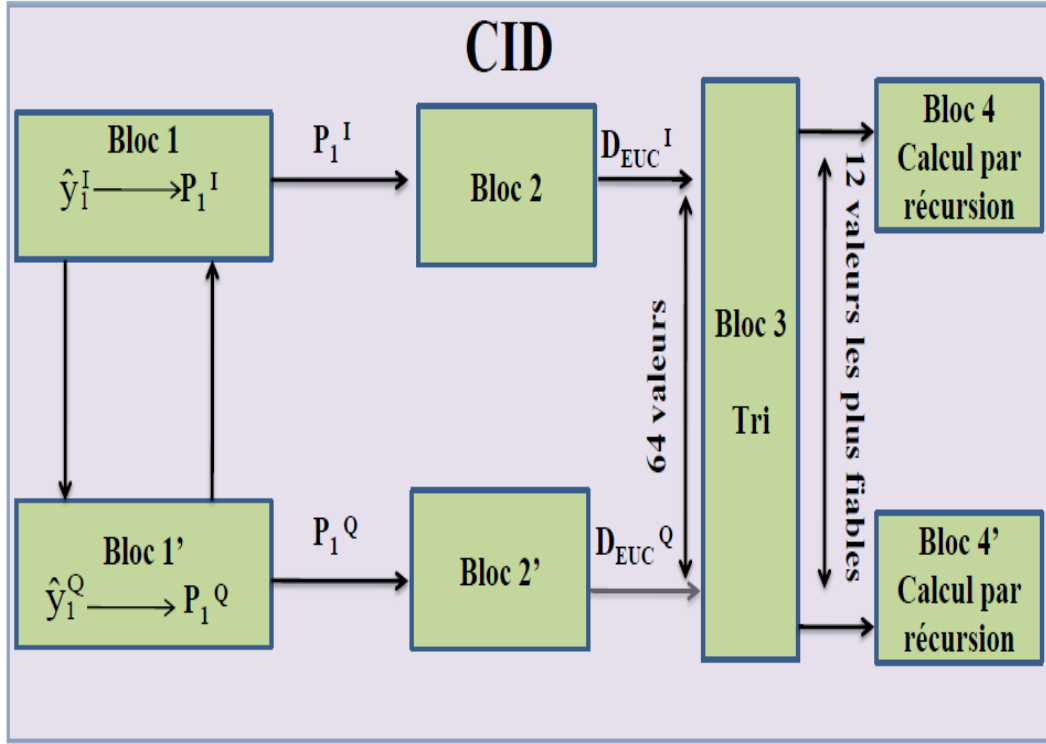


FIGURE 4.5 – Architecture du bloc Calcul Initial des Distances euclidiennes (CID)

pas dupliqué le bloc 3 car nous ne pouvons pas trier les parties réelles indépendamment des parties imaginaires pour les distances euclidiennes.

Plus précisément, le bloc 1 (resp. 1') détecte le vecteur P_1^I (resp. P_1^Q) du symbole candidat S_{2i} le plus proche pour chaque élément fixé $S_{2i-1} = \alpha_k$, avec $\{k = 0, 1, \dots, 63\}$. Le bloc 2 (resp. 2') calcule la coordonnée réelle (resp. imaginaire), D_{EUC}^I (resp. D_{EUC}^Q), de la distance euclidienne du symbole détecté par le bloc 1 (resp. 1') sachant que $S_{2i} = \alpha_k$, avec $\{k = 0, 1, \dots, 63\}$. Ensuite, le bloc 3 reçoit séquentiellement les parties réelles et imaginaires des 64 valeurs de distance euclidienne. Ces valeurs représentent les distances euclidiennes associées au symbole S_{2i} le plus proche pour chaque élément fixé $S_{2i-1} = \alpha_k$. Le bloc 3 a pour but de classer les valeurs des symboles du vecteur détecté par ordre croissant en fonction de leur fiabilité respective. Finalement, le bloc 4 (resp. 4') calcule tous les points voisins des valeurs triées. Pour ce faire, la méthode de calcul récursif est appliquée en utilisant la technique de sous-régions dynamiques $\{2 \times 12 + 4 \times 8 + 6 \times 6 + 52 \times 4\}$ comme expliquée dans la section 3.2.2.

4.3.1.1 Bloc 1 - $\{\widehat{y_1^{(I)}} \rightarrow P_1^I\}$

Le rôle principal de ce bloc est de détecter le vecteur P_1^I du symbole candidat S_{2i} le plus proche pour chaque élément fixé $S_{2i-1} = \alpha_k$, avec $\{k = 0, 1, \dots, 63\}$. L'expression de P_1^I est donnée dans l'équation 3.14 du chapitre 3 :

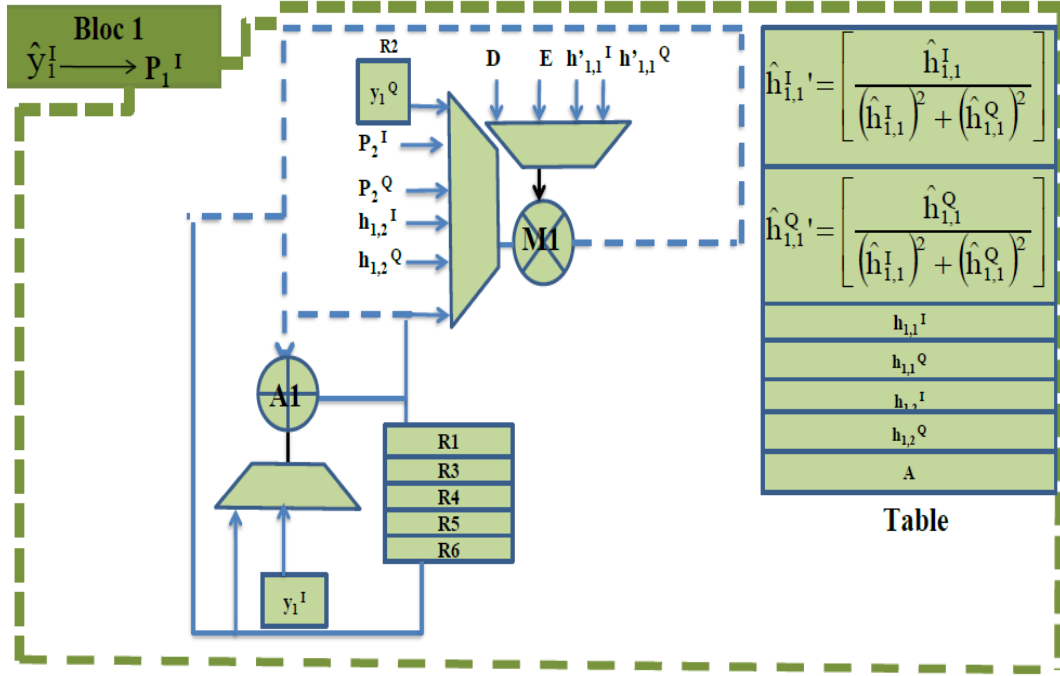


FIGURE 4.6 – Représentation architecturale du bloc 1 - $\{\widehat{y}_1^{(I)} \rightarrow P_1^I\}$

$$\begin{aligned}
 P^{I_1} = & \overbrace{\left(\widehat{h_{1,1}^{(I)}}' \cdot \widehat{y_1^{(I)}} + \widehat{h_{1,1}^{(Q)}}' \cdot \widehat{y_1^{(Q)}} \right)}^C - \overbrace{\left(\widehat{h_{1,1}^{(I)}}' \cdot \widehat{h_{1,2}^{(I)}} + \widehat{h_{1,1}^{(Q)}}' \cdot \widehat{h_{1,2}^{(Q)}} \right)}^D \cdot P^{I_2} \\
 & + \overbrace{\left(\widehat{h_{1,1}^{(I)}}' \cdot \widehat{h_{1,2}^{(Q)}} - \widehat{h_{1,1}^{(Q)}}' \cdot \widehat{h_{1,2}^{(I)}} \right)}^E \cdot P^{Q_2}
 \end{aligned} \tag{4.1}$$

Il est à noter que les coefficients du canal ainsi que le terme A, qui est utilisé pour le calcul $\widehat{y_1^{(I)}} = y_1^{(I)} - A$, sont supposés être parfaitement connus par le détecteur. Ils sont donc stockés dans une table.

A partir de l'équation 3.14, nous devons calculer les termes $\widehat{y_1^{(I)}}$, C, D, et E afin de finalement déterminer P_1^I . Pour ce faire, le bloc $\widehat{y_1^{(I)}} \rightarrow P_1^I$ détaillé dans la figure 4.6 est une architecture adaptative permettant de calculer les quatres termes précédents ainsi que le terme final P_1^I .

L'architecture proposée se compose de trois multiplexeurs, un additionneur réel, un multiplieur réel, six registres et une table. Rappelons qu'un cycle d'exécution correspond au temps de traverser de deux multiplieurs et d'un additionneur de deux nombres réels. Nous avons défini cette architecture de telle sorte que les ressources arithmétiques soient utilisées durant chaque cycle d'exécution.

La figure 4.7 illustre le calcul du terme C par le bloc $\widehat{y_1^{(I)}} \rightarrow P_1^I$. Ce calcul nécessite 2 cycles d'exécution. Un cycle utilise le multiplieur M1 afin de calculer $\{\widehat{h_{1,1}^{(I)}}' \cdot \widehat{y_1^{(I)}}\}$. Le second cycle utilise le multiplieur M1 et l'additionneur A1 pour le reste. La valeur du terme C est mémorisée dans le

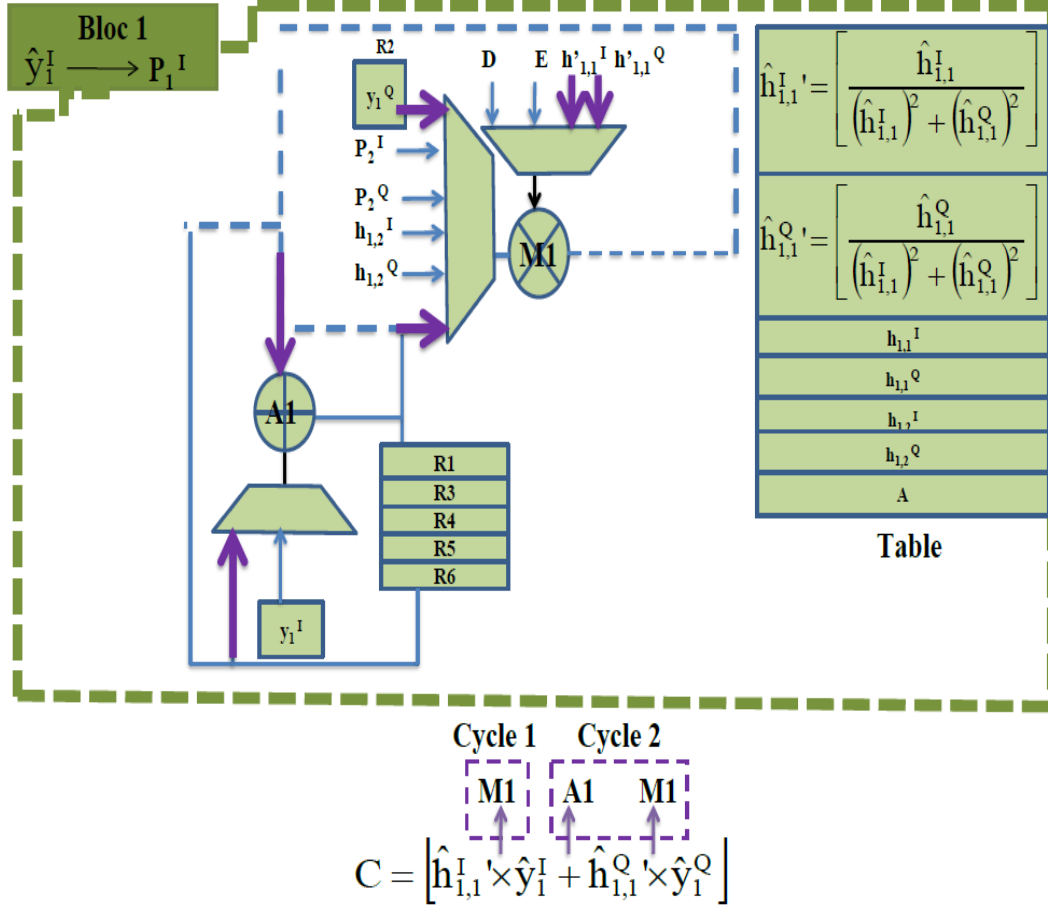


FIGURE 4.7 – Architecture du bloc 1 - $\{\hat{y}_1^{(I)} \rightarrow P_1^I\}$ pour le calcul du terme C

registre R3. Le même processus est répété séquentiellement au sein du bloc 1 pour calculer D et E qui sont eux stockés dans les registres R4 et R5. En parallèle, le bloc 1' calcule les termes F, G et H afin de déterminer le terme P_1^Q . Il est à noter que les termes $\{C, D, E, F, G \text{ et } H\}$ sont constants pour calculer l'ensemble de 64 valeurs composants les vecteurs P_1^I et P_1^Q .

Nous abordons maintenant le calcul du terme P_1^I . Ce calcul est illustré dans la figure 4.8. Notre but est de détecter le terme P_1^I pour chaque symbole candidat composé de P_2^I et P_2^Q . Seules 8 valeurs sont possibles pour P_2^I et P_2^Q . C'est pourquoi, nous faisons le calcul pour les 8 valeurs de P_2^I en fixant P_2^Q . Tout d'abord, nous avons besoin d'un cycle pour calculer le terme CEP pour une valeur fixée de P_2^Q en utilisant l'additionneur réel A1 et le multiplieur réel M1. La valeur calculée est mémorisée dans le registre R6. Ensuite, il nous faut un deuxième cycle pour le calcul du reste de l'équation qui dépend de la valeur du P_2^I . Pour les sept autres valeurs du P_2^I , un cycle est nécessaire par valeur en exploitant la valeur initiale stockée dans le registre R6.

Au total, 9 cycles d'exécution sont nécessaires pour chaque P_2^Q comme illustré dans la figure 4.9. Au final, afin de détecter les 64 valeurs du vecteur P_1^I , nous avons besoin de 79 cycles comme indiqué dans la figure 4.9.

L'efficacité de cette architecture doit être illustrée. C'est pourquoi, des diagrammes de Gantt qui explicitent l'utilisation de l'additionneur réel, du multiplieur réel et l'occupation des six registres du

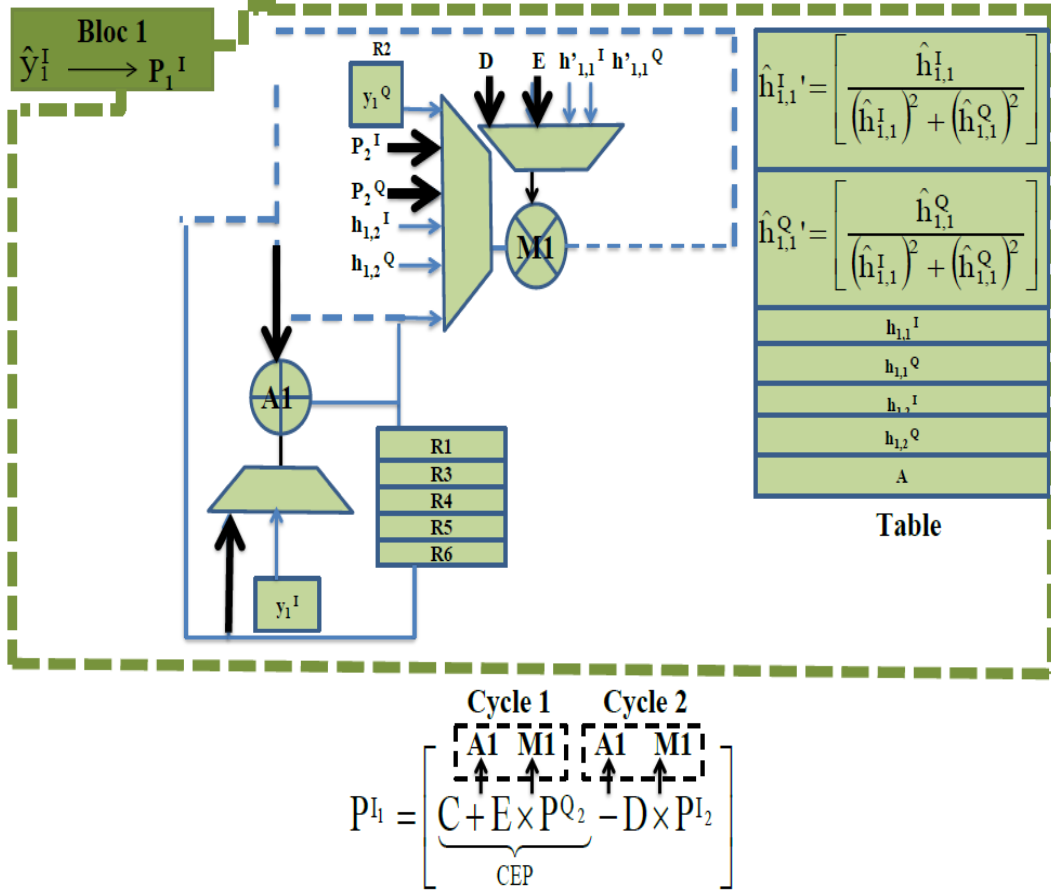


FIGURE 4.8 – Architecture du bloc 1 - $\{\widehat{y_1^{(I)}} \rightarrow P_1^I\}$ pour calculer P_1^I

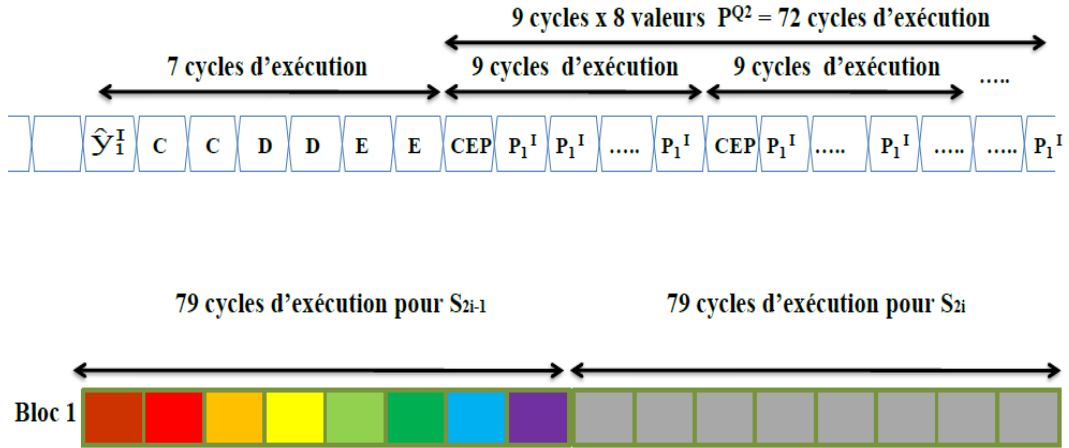


FIGURE 4.9 – Organisation temporelle du bloc 1 - $\{\widehat{y_1^{(I)}} \rightarrow P_1^I\}$

bloc $\widehat{y_1^{(I)}} \rightarrow P_1^I$ sont présentés dans la figure 4.10. Cette figure démontre que l'additionneur réel A1 et le multiplieur réel M1 sont presque toujours assignés au cours du traitement. De même, les six registres du bloc 1 sont le plus souvent occupés. Le bloc $\widehat{y_1^{(I)}} \rightarrow P_1^I$ nécessite 9 cycles d'exécution à l'issue desquels le premier terme P_1^I de la première distance euclidienne est disponible. Ce terme est envoyé directement au bloc 2 afin de calculer la première distance euclidienne D_{EUC}^I .

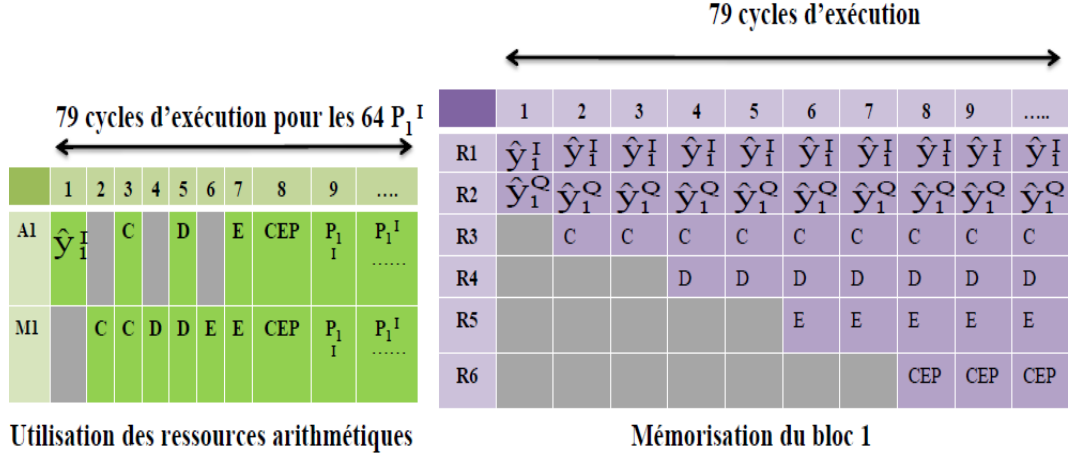


FIGURE 4.10 – Utilisation des ressources arithmétiques et de mémorisation du bloc 1 - $\{\widehat{y}_1^{(I)} \rightarrow P_1^I\}$

4.3.1.2 Bloc 2 - $\{D_{EUC}^I\}$

Le rôle du bloc 2 (resp. bloc 2') est de calculer la coordonnée réelle D_{EUC}^I (resp. imaginaire D_{EUC}^Q) de la distance euclidienne de chaque symbole détecté P_1^I (resp. $\widehat{y}_1^{(I)} \rightarrow P_1^I$) (resp. $\{B \rightarrow P_1^Q\}$). Ces distances seront les valeurs initiales pour les calcul effectués au sein des blocs 4 et 4' pour toutes les distances euclidiennes des points voisins en appliquant un calcul récursif.

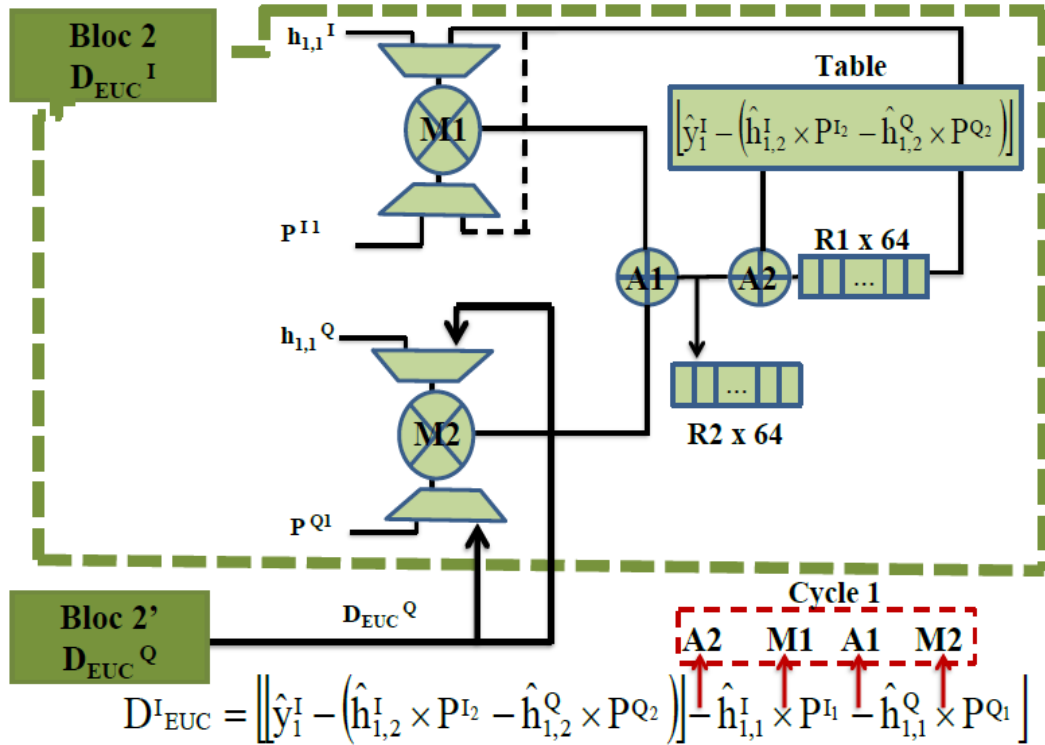


FIGURE 4.11 – Découpe architecturale du bloc 2 - $\{D_{EUC}^I\}$

Comme les 8 valeurs possibles pour P_2^I et P_2^Q sont déjà fixées, nous proposons de stocker les expressions dépendantes de P_2^I et de P_2^Q dans une table. Ainsi, il ne reste plus qu'à calculer les expressions de P_1^I et de P_1^Q .

L'architecture du bloc 2, décrite dans la figure 4.11, est composée de quatre multiplexeurs, deux additionneurs réels, deux multiplieurs réels, deux registres de 64 valeurs et une table. A partir de cette architecture et de l'équation 3.19, nous estimons que le calcul du terme D_{EUC}^I nécessite qu'un seul cycle d'exécution. Un second cycle est nécessaire pour calculer le terme au carré $\{D_{EUC}^2 = D_{EUC}^I{}^2 + D_{EUC}^Q{}^2\}$.

Au cours du premier cycle d'exécution, nous utilisons les deux multiplieurs réels $M1$ et $M2$ associés respectivement aux deux additionneurs réels $A1$ et $A2$ afin de calculer le terme D_{EUC}^I . Cette valeur est stockée dans le registre $R1$. Puis, à partir du terme D_{EUC}^Q du bloc 2', nous effectuons le calcul de l'expression $\{D_{EUC}^2 = D_{EUC}^I{}^2 + D_{EUC}^Q{}^2\}$ au cours du second cycle d'exécution en assignant $M1$, $M2$ et $A1$. L'expression résultante correspond à la première distance euclidienne D_{EUC}^2 . Elle est enregistrée dans le registre $R2$. Le même processus est répété séquentiellement au sein du bloc 2 pour calculer toutes les autres distances euclidiennes.

Ainsi, deux cycles d'exécution sont nécessaires pour calculer la valeur d'une distance euclidienne. Le premier cycle d'exécution pour calculer la valeur D_{EUC}^I . Le second cycle pour calculer le terme au carré $\{D_{EUC}^2 = D_{EUC}^I{}^2 + D_{EUC}^Q{}^2\}$. Comme nous l'avons déjà expliqué dans le chapitre précédent, le vecteur à deux dimensions du premier symbole candidat S_{2i-1} comprend 64 valeurs de distance euclidienne sur sa première dimension. Ces valeurs représentent les distances euclidiennes associées au symbole S_{2i} le plus proche pour chaque élément fixé $S_{2i-1} = \alpha_k$. Ainsi, 128 cycles d'exécution sont nécessaires pour calculer toutes les valeurs de la première dimension de ce vecteur.

De plus, une organisation temporelle des blocs 1 et 2 est proposée, comme illustré la figure 4.12. Cette organisation montre l'ordonnancement des différentes opérations réalisées par l'ensemble des blocs 1 et 2. Le bloc 1 nécessite 9 cycles d'exécution à l'issue desquels le premier symbole détecté P_1^I est disponible. Après ces 9 cycles d'exécution, le bloc 2 commence à calculer la première distance euclidienne qui nécessite en particulier 2 cycles d'exécution. Par conséquent, le bloc 1 nécessite 79 cycles d'exécutions pour détecter les 64 symboles candidats du vecteur P_1^I . Le bloc 2 nécessite 128 cycles d'exécutions pour calculer les 64 valeurs des distances euclidienne associées, effectués quasi

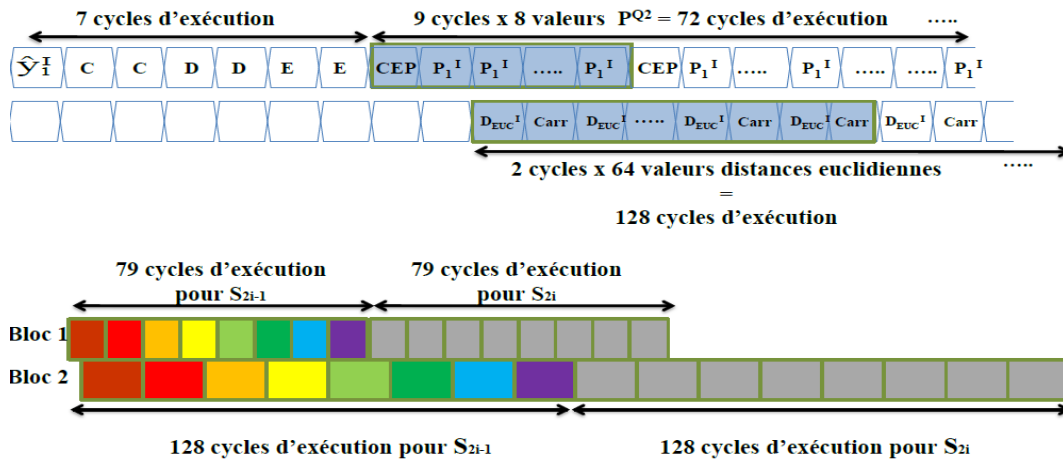


FIGURE 4.12 – Organisation temporelle du bloc 2 - $\{D_{EUC}^I\}$

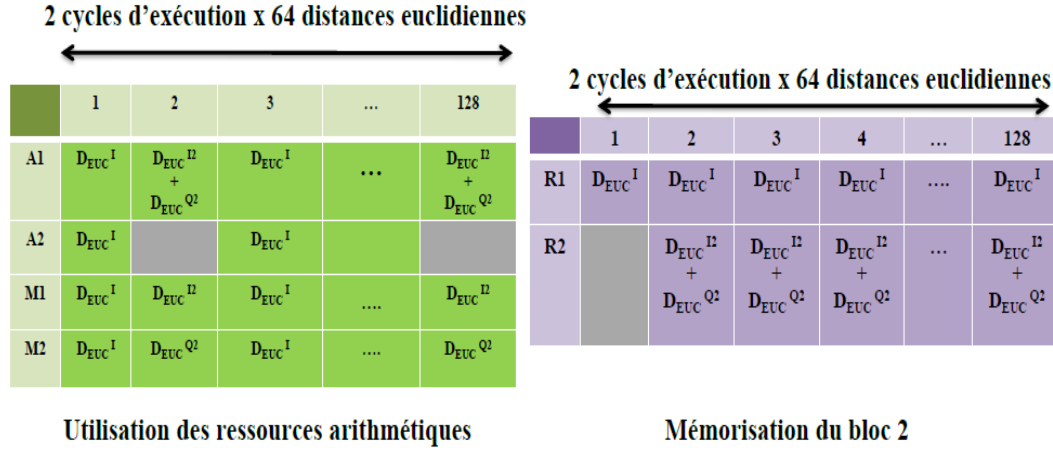


FIGURE 4.13 – Utilisation des ressources arithmétiques et de mémorisation du bloc 2 - $\{D_{EUC}^I\}$

simultanément en parallèle avec le bloc 1.

Des diagrammes de Gantt qui explicitent l'utilisation des deux additionneurs réels et des deux multiplieurs réels et l'occupation des deux registres de 64 valeurs du bloc 2 sont présentés dans la figure 4.13. Cela montre que les additionneurs réels A1 et A2 et les multiplieurs réels M1 et M2 sont presque toujours assignés au cours du traitement. De même, les deux registres de 64 valeurs dans le bloc 2 sont progressivement assignés.

4.3.1.3 Bloc 3 - Tri

Le bloc de tri est le troisième sous-bloc du bloc principal CID. Le rôle principal de ce bloc est de trouver le nombre de distances euclidiennes à calculer sur la dimension S_{2i} pour chaque valeur fixée $\{S_{2i-1} = \alpha_k\}$, avec $k = 0, 1, \dots, 63$, comme expliqué dans la section 3.2.2. Pour ce faire, un tri des distances euclidiennes du vecteur S_{2i-1} est appliqué pour classer ces distances selon leur fiabilité sur la première dimension. Par exemple, pour les deux premières distances les plus fiables, nous chercherons au sein du quatrième bloc 12 points voisins par récursion sur la deuxième dimension.

Comme expliqué dans la section 3.2.2, nous avons besoin de trouver les 12 distances euclidiennes les plus fiables sur la première dimension, afin de fixer leur intervalle respectif sur la deuxième dimension selon leur fiabilité respective. Pour les $\{64-12\}$ distances euclidiennes les moins fiables parmi les 64 valeurs des distances euclidiennes calculées par le bloc 2, nous avons fixé un intervalle de 4 points voisins, comme justifié dans le chapitre 3.

Nous proposons une architecture pour le bloc 3 qui se compose de douze comparateurs, un transcodeur et de 24 registres comme illustré dans la figure 4.14. Tous les deux cycles d'exécution, le bloc 3 reçoit une nouvelle valeur de distance euclidienne calculée par le bloc 2. Grâce aux 12 comparateurs et un transcodeur, le bloc 3 compare cette distance euclidienne à celles des douze distances euclidiennes les plus fiables. Le cas échéant, il va la stocker dans un des registres $\{Rx, \text{ avec } x = 1, 2, \dots, 12\}$. En résumé, ce bloc reçoit 64 distances euclidiennes, il va trier les 12 distances euclidiennes parmi 64 par ordre de fiabilité puis les mémoriser dans les registres $\{Rx', \text{ avec } x' = 1', 2', \dots, 12'\}$.

L'organisation temporelle du bloc 3 par rapport aux blocs 1 et 2 est donnée dans la figure 4.15. Le bloc 3 commence le tri dès qu'il reçoit la première distance euclidienne provenant du bloc 2. Ainsi, tous les 2 cycles d'exécution le bloc 3 reçoit une nouvelle valeur. Cependant, un seul

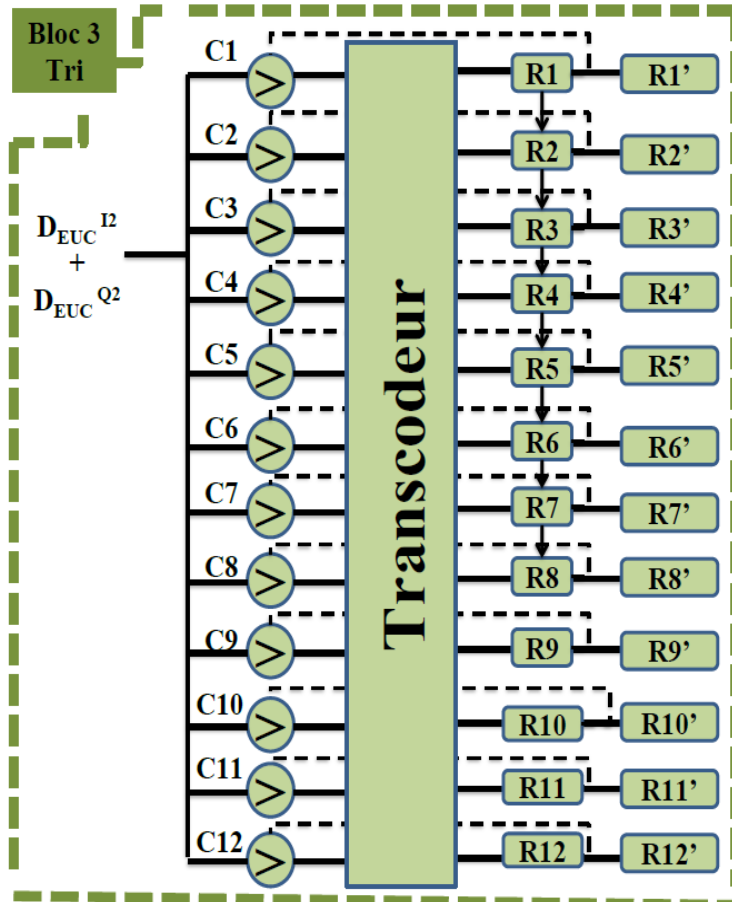


FIGURE 4.14 – Découpe architecturale du bloc 3 - Tri

cycle d'exécution est nécessaire pour la comparer aux 12 valeurs stockées dans les 12 registres du bloc 3.

Par conséquent, pour trier 12 valeurs parmi les 64 distances euclidiennes les plus fiables, 128 cycles d'exécution sont nécessaires comme illustré la figure 4.15. Ces 128 cycles d'exécution sont effectués avec un décalage de 2 cycles, par rapport à l'exécution du bloc 2.

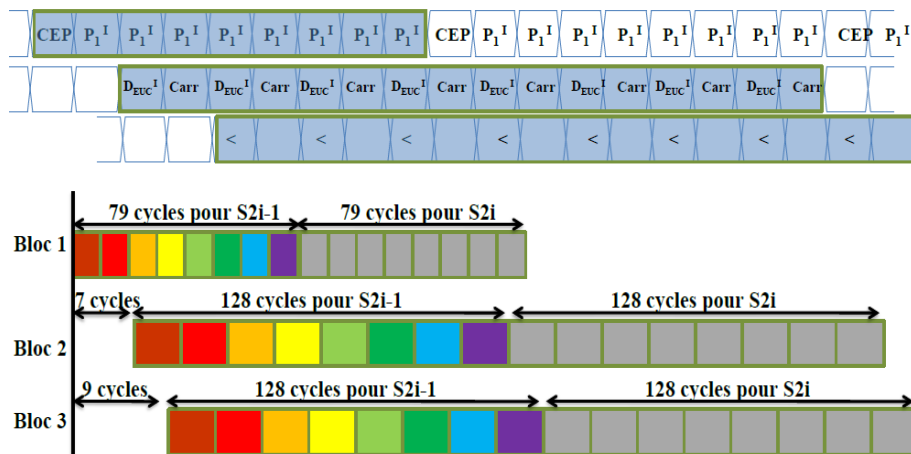


FIGURE 4.15 – Organisation temporelle du bloc 3 - Tri

4.3.1.4 Bloc 4 - Calcul récursif

Une fois le tri effectué par le bloc 3, le bloc 4 reçoit les 12 valeurs des distances euclidiennes les plus fiables selon la première dimension $\{S_{2i-1}\}$, triées par ordre de fiabilité. Le bloc 4 a pour fonctionnalité de calculer récursivement pour chaque valeur calculée selon la dimension $\{S_{2i-1}\}$ un nombre de distances euclidiennes selon la deuxième dimension $\{S_{2i}\}$. Ce nombre est fixé en fonction de l'ordre de fiabilité des symboles comme expliqué dans la section 3.2.2.

Le bloc 4 doit chercher 12 points par récursion pour les deux premières distances euclidiennes les plus fiables. Pour les quatre suivantes, 8 points voisins seront recherchés par récursion. Ensuite, pour les six distances euclidiennes suivantes, 6 points voisins seront recherchés par récursion. Enfin, pour les autres distances euclidiennes $\{64 - 12\}$, 4 points voisins seront calculés par récursion. Ainsi, le bloc 4 calcule les points voisins des valeurs triées par le bloc 3 selon le jeu de paramètres $\{2 \times 12 + 4 \times 8 + 6 \times 6 + 52 \times 4\}$, comme expliqué dans la section 3.2.2. Rappelons que la méthode de calcul par récursion consiste tout d'abord à additionner le terme $\{2 \times h_{j,i}^I\}$ (resp. $\{2 \times h_{j,i}^Q\}$) à la distance euclidienne du point précédent. Puis, une multiplication est nécessaire pour l'élévation au carré du terme obtenu $\{DEUC^I\}$ (resp. $\{DEUC^Q\}$).

Pour ce faire, nous proposons une architecture composée de trois multiplexeurs, de deux additionneurs réels, d'un multiplieur réel et d'un registre comme illustré dans la figure 4.16. Avec cette architecture, nous pouvons calculer en un cycle d'exécution une nouvelle distance euclidienne. Cela s'effectue par le passage par l'additionneur A1 pour sommer le terme $\{2 \times h_{j,i}^I\}$ (resp. $\{2 \times h_{j,i}^Q\}$) à la distance euclidienne du point précédent $\{DEUC^I\}$. Puis, nous utilisons le multiplieur pour l'élévation au carré $\{DEUC^{I2}\}$ (resp. $\{DEUC^Q\}$). Finalement, le terme modifié $\{DEUC^{I2}\}$ (resp. $\{DEUC^Q\}$) est sommé avec le terme constant $\{DEUC^{Q2}\}$ (resp. $\{DEUC^I\}$) pour obtenir la nouvelle distance euclidienne d'un nouveau point.

A partir du jeu de paramètres $\{2 \times 12 + 4 \times 8 + 6 \times 6 + 52 \times 4\}$, nous devons calculer au total 300 distances euclidiennes. Chaque recherche nécessite un cycle d'exécution. Nous avons donc besoin de 300 cycles d'exécution afin de déterminer toutes les distances euclidienne requises. Afin de

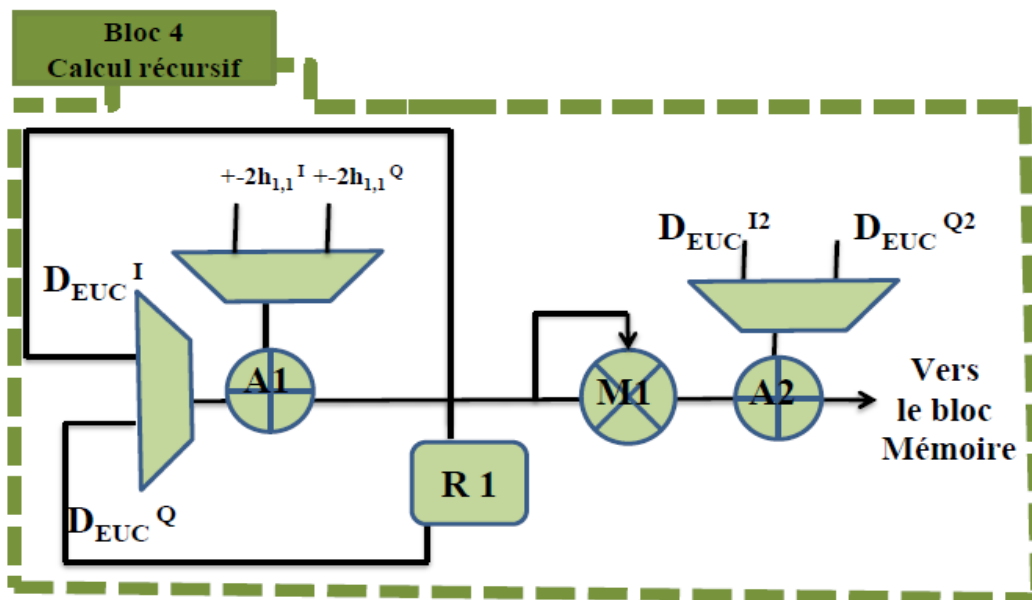


FIGURE 4.16 – Découpe architecturale du bloc 4 - Calcul récursif

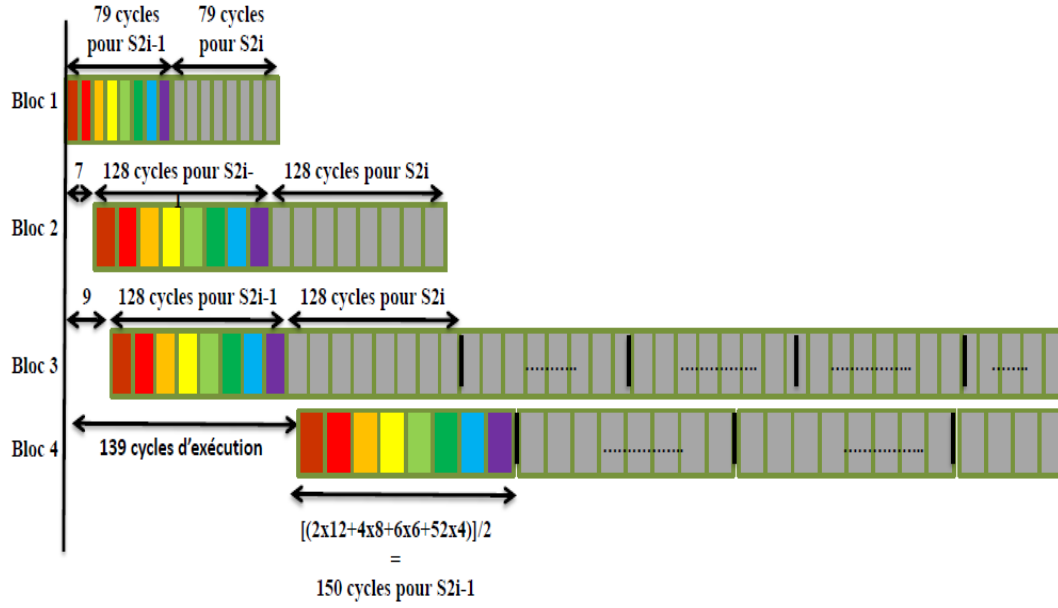


FIGURE 4.17 – Organisation temporelle du bloc 4 - Calcul récursif

réduire la latence de ce bloc, nous proposons de calculer les 300 distances euclidiennes à l'aide de deux blocs 4 et 4'. Les deux blocs ont une architecture similaire. Ainsi, chaque bloc doit calculer 150 distances euclidiennes. Ces distances sont transmises directement au bloc mémoire afin de les mettre à jour lors de la deuxième inter-itération par les informations extrinsèques provenant du décodeur.

Dans la figure 4.17, nous présentons l'organisation temporelle de tous les sous-blocs constituant le bloc CID. Nous remarquons que les trois premiers sous-blocs peuvent travailler conjointement. En revanche, le bloc 4 nécessite la fin du tri réalisé au sein du bloc 3. Cela nécessite 139 cycles

150 cycles d'exécution							
	1	2	...	8	9	...	150
A1	$-2h_{1,1}^Q + D_{EUC}^Q$	$-2h_{1,1}^Q + D_{EUC}^Q$...	$-2h_{1,1}^I + D_{EUC}^I$	$-2h_{1,1}^I + D_{EUC}^I$
A2	$(D_{EUC}^Q)^2 + (D_{EUC}^I)^2$	$(D_{EUC}^Q)^2 + (D_{EUC}^I)^2$...	$(D_{EUC}^I)^2 + (D_{EUC}^Q)^2$	$(D_{EUC}^I)^2 + (D_{EUC}^Q)^2$
M1	$(D_{EUC}^Q)^2$	$(D_{EUC}^Q)^2$...	$(D_{EUC}^I)^2$	$(D_{EUC}^I)^2$
Assignation des ressources arithmétiques							
150 cycles d'exécution							
	1	2	..	8	9	..	150
R1	$-2h_{1,1}^Q + D_{EUC}^Q$	$-2h_{1,1}^Q + D_{EUC}^Q$..	$2h_{1,1}^I + D_{EUC}^I$	$2h_{1,1}^I + D_{EUC}^I$
Mémorisation du bloc 4							

FIGURE 4.18 – Assignation des ressources arithmétiques et de mémorisation pour le bloc 4 - Calcul récursif

d'exécution. Ce nombre correspond au temps de traitement du bloc 3. Puis, le bloc 4 commence à calculer la moitié des distances euclidiennes tandis que le bloc 4' calcule en parallèle les autres. Cela nécessite 150 cycles d'exécution supplémentaires.

Au final, l'exécution du bloc CID nécessite $\{139+150=289\}$ cycles d'exécution pour effectuer le calcul initial de toutes les distances euclidiennes requises par l'algorithme de détection basé sur la propagation de croyance.

Rappelons que le calcul au sein du bloc CID est uniquement nécessaire lors de la première inter-itération entre le détecteur et le décodeur. Dès la deuxième inter-itération, une mise à jour des informations enregistrées dans le bloc mémoire sera fait par les informations extrinsèques provenant du décodeur lors de l'itération précédente. La figure 4.18 présente les diagrammes de Gantt des deux additionneurs réels A1 et A2 et du multiplieur réel M1 et l'occupation du registre R1 du bloc 4. Nous remarquons que les additionneurs réels A1 et A2 et le multiplieur M1 sont toujours assignés au cours du traitement. C'est également le cas pour le registre R1.

4.3.2 Bloc Mémoire

Le rôle de ce bloc est de sauvegarder les 64 vecteurs LRV du symbole candidat S_{2i-1} calculés par le bloc CID. Ces 64 vecteurs du symbole S_{2i-1} sont construits en fonction de leur deuxième dimension S_{2i} par la méthode des régions dynamiques $\{2 \times 12 + 4 \times 8 + 6 \times 6 + 52 \times 4\}$, comme expliqué dans le chapitre 3. Le bloc mémoire est construit à partir de 64 vecteurs S_{2i-1} où chaque vecteur est composé de 64 valeurs associées à S_{2i} , comme illustré dans la figure 4.19. Au final, la mémoire comprend 64^2 cases.

Chaque vecteur du bloc mémoire est alimenté par les valeurs calculées au sein du bloc CID sachant que leur nombre n_{omb} est spécifié par la méthode des régions dynamiques. Les $(64 - n_{omb})$ cases qui restent seront saturées par la valeur la moins fiable des valeurs calculées pour le vecteur associé. Ce choix a été validé par des simulations de performances présentées dans la section 3.3. Il est à noter que ce bloc doit transmettre directement les valeurs calculées par le bloc CID aux nœuds des

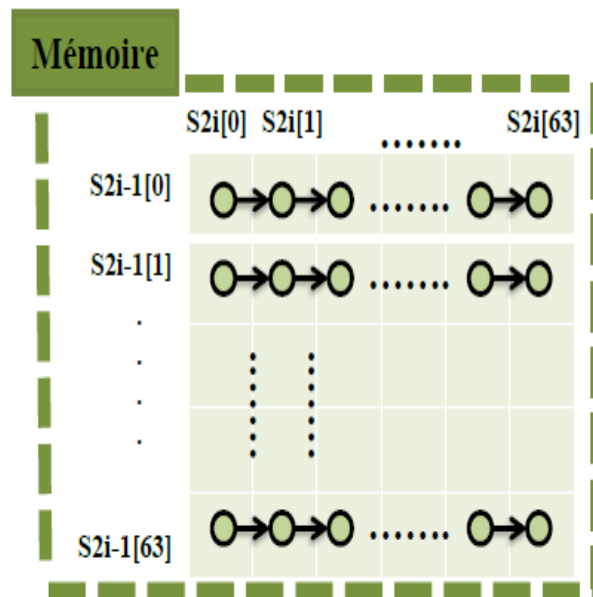


FIGURE 4.19 – Bloc Mémoire

symboles candidats, S_{2i-1} et S_{2i} avec $i = 1, 2, \dots, 192$, sans passer par le bloc AT (Additionneur-Tri) lors de la première inter-itération. Dès la deuxième inter-itération, le bloc mémoire se met à jour via le bloc AT (Additionneur-Tri) à partir des informations extrinsèques provenant du décodeur lors de l'itération précédente.

4.3.3 Bloc AT (Additionneur-Tri)

Ce bloc est utilisé dès la deuxième inter-itération entre le détecteur et le décodeur. Son rôle principal est de mettre à jour les informations sauvegardées dans la mémoire à partir des informations provenant du décodeur lors de l'itération précédente. Pour ce faire, il exploite les 8 informations extrinsèques (de $S_{2i}[n_{m0}]$ à $S_{2i}[n_{m7}]$) les plus fiables sur la deuxième dimension S_{2i} provenant du décodeur. Ces informations extrinsèques sont obtenues après la mise à jour dans le calcul courant des symboles candidats par les informations *a priori* des autres symboles candidats. Ensuite, il additionne les informations extrinsèques reçues à celles qui sont stockées dans la mémoire selon le principe décrit dans la section 3.3.

Nous rappelons que le sous-ensemble de valeurs devant être mises à jour pour le symbole S_{2i-1} est noté $\{S_{2i-1}^{upd} = dec_{nv} + det_{nc}\}$. $dec_{nv} = 2$ est la contribution provenant du décodeur LDPC non-binaire. $det_{nc} = 2$ est la contribution provenant du détecteur MIMO. Ainsi, quatre vecteurs du symbole S_{2i-1} seront mis à jour par les 8 informations extrinsèques les plus fiables selon la deuxième dimension S_{2i} . En outre, afin de compenser les messages tronqués pour la première dimension S_{2i-1} , nous ne mettons à jour qu'une valeur pour chacun des $(64-4)$ vecteurs restants.

Pour ce faire, deux multiplexeurs sont nécessaires dans l'architecture du bloc AT qui est proposée dans la figure 4.20. Par multiplexage, nous additionnons séquentiellement les 4 vecteurs du symbole

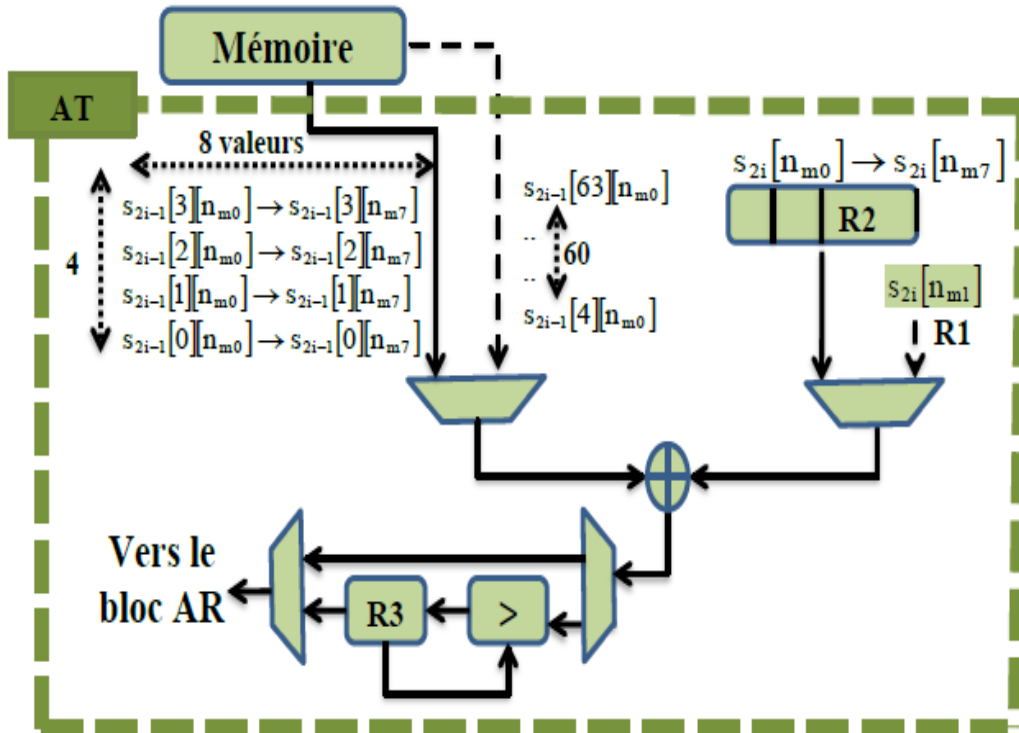


FIGURE 4.20 – Découpe architecturale du bloc AT

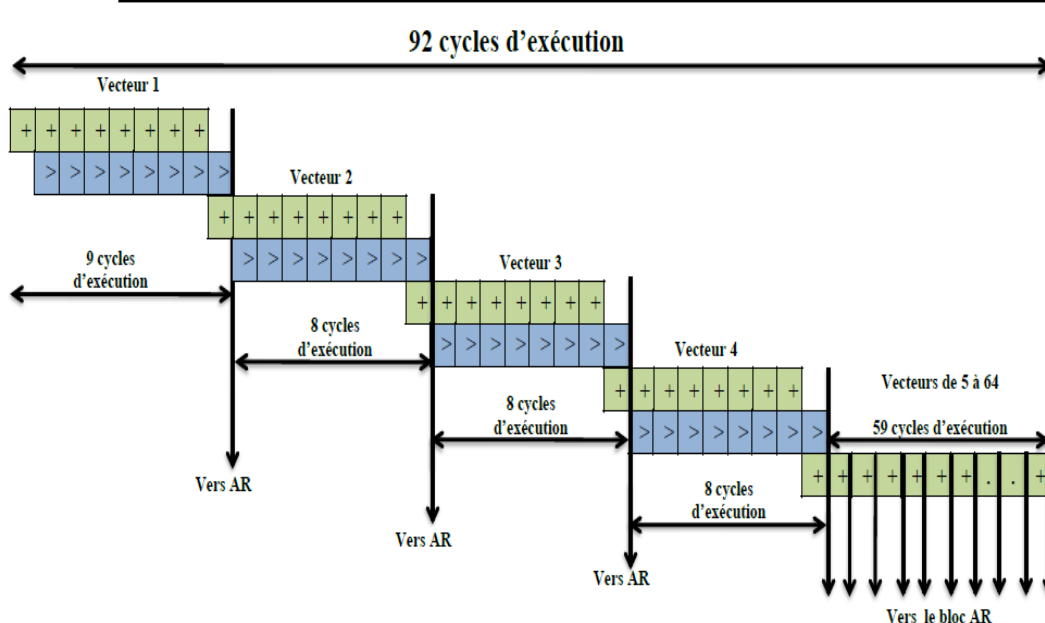


FIGURE 4.21 – Organisation temporelle pour le bloc AT

S_{2i-1} composés chacun de 8 valeurs aux vecteurs contenant les informations extrinsèques. Pour chaque vecteur traité, un tri est nécessaire. Un comparateur associé à un registre R3 est nécessaire pour trouver l'information la plus fiable pour chaque vecteur traité. A la fin du traitement de chaque vecteur, la valeur la plus fiable est sauvegardée dans un registre de 64 valeurs. Ce traitement est appliqué séquentiellement aux quatre différents vecteurs.

Pour les autres vecteurs (64-4), nous allons additionner la valeur la plus fiable $S_{2i}[n_{m0}]$ provenant du décodeur du symbole S_{2i} à la valeur qui est associée à chaque vecteur. Ainsi, la mise à jour sera directement transmise au bloc AR sans passer par le comparateur et le registre R3. Finalement, un vecteur composé de 64 valeurs S_{2i-1} contenant les valeurs les plus fiables sera construit au sein du bloc AR.

Dans la figure 4.21, nous détaillons l'organisation temporelle du bloc AT. Comme expliqué précédemment, nous traitons séquentiellement les 64 vecteurs du symbole S_{2i-1} . Tout d'abord, 9 cycles d'exécution sont nécessaires pour déterminer l'information la plus fiable du premier vecteur associé au symbole S_{2i-1} en effectuant des additions et des comparaisons. De même, 8 cycles d'exécution sont nécessaires pour les 3 vecteurs suivants.

Pour les 60 vecteurs restants, nous avons besoin d'effectuer qu'une addition à chaque fois. Ainsi, 92 cycles d'exécutions sont au total nécessaires pour mettre à jour les informations sauvegardées dans la mémoire par les informations extrinsèques provenant du décodeur.

4.3.4 Bloc AR (Additionneur-Registre)

Les calculs au sein des nœuds associés aux symboles candidats sont effectués à l'aide d'un additionneur et de deux registres de 64 valeurs, comme illustré dans la figure 4.22. Ils permettent de tenir compte des informations *a priori* pour les autres symboles reçus. Ainsi, les informations associées aux symboles candidats sont mises à jour à partir des informations provenant des autres

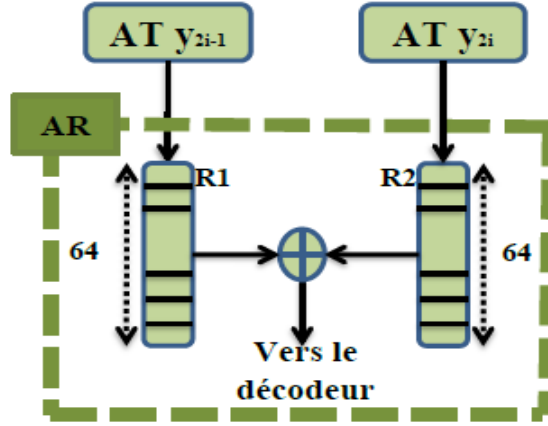


FIGURE 4.22 – Architecture du bloc AR

symboles reçus et des informations extrinsèques provenant du décodeur par le bloc AT lors. Les informations souples associées aux symboles sont ensuite transmises au décodeur.

4.4 Architecture du décodeur LDPC non-binaire

Dans cette section, nous allons présenter l'architecture du décodeur LDPC non-binaire. Notons que cette architecture s'inspire fortement de celle décrite dans le cadre du projet Davinci [82], [83]. La différence majeure concerne le séquençement. Dans notre contexte, ce dernier est vertical ayant une convergence plus rapide et non horizontal comme dans les travaux de référence. Ces travaux ont abouti au développement d'une architecture efficace d'un décodeur LDPC non-binaire basé sur des matrices de parités ultra-creuses.

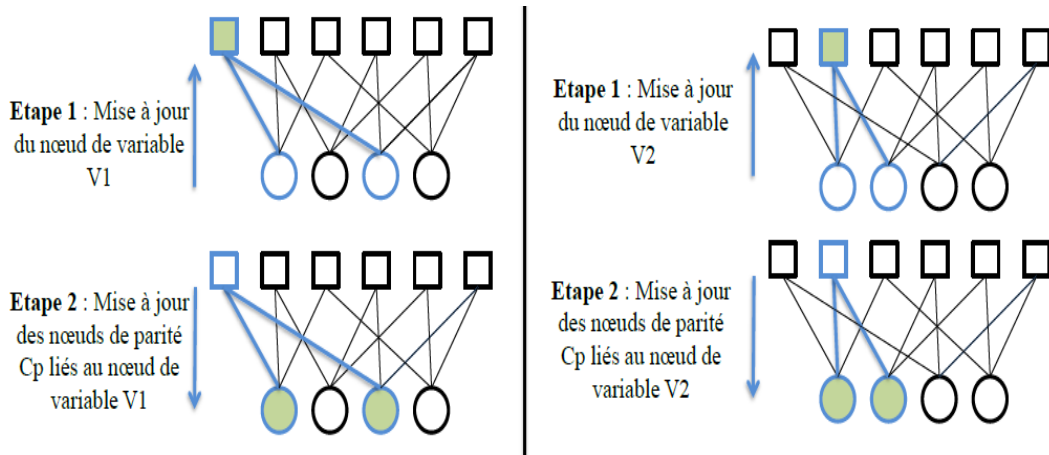


FIGURE 4.23 – Illustration des étapes d'une intra-itération de décodage selon le séquençement *shuffle* à permutation verticale

4.4.1 Architecture globale du décodeur et son séquençement

Dans la suite de cette section, nous allons considérer le décodeur LDPC non binaire avec un rendement égal à $R = 1/2$ et pour une longueur de trame $N = 384$. Cette trame est composée de symboles basés sur $GF(64)$. Ces caractéristiques ont été choisies de sorte à pouvoir comparer le code proposé à celui adopté par la norme IEEE WiMAX [60] que nous prenons pour référence. Ainsi, nous pourrions faire plus facilement des comparaisons. De plus, nous avons choisi la matrice de parité qui a un degré de connectivité $d_v = 2$ pour chaque nœud de variable et $d_c = 4$ pour chaque nœud de parité afin d'avoir de bonnes performances et une architecture de décodage régulière comme expliqué dans [45],[96].

L'architecture proposée dans le cadre de notre étude est une architecture séquentielle avec un séquençement *shuffle* à permutation verticale. Le principe de cet ordonnancement est illustré dans la figure 4.23. Il consiste à traiter les nœuds de variable un à un lors de chaque inter-itération entre le détecteur MIMO et le décodeur LDPC non binaire. Ce séquençement est différent du séquençement classique "*flooding*" qui traite tous les nœuds de parité puis tous les nœuds de variable, et ainsi de suite de manière itérative. L'intérêt majeur du séquençement *shuffle* vertical est de favoriser la convergence [67] au sein d'un récepteur itératif. Une grande partie des nœuds de variable peut ainsi profiter de la mise à jour des nœuds traités préalablement. Cela est possible grâce aux connexions communes via les nœuds de parité du graphe bipartite. Un autre avantage de ce type de séquençement est une diminution significative de la latence d'exécution pour des performances équivalentes, comme expliqué dans le deuxième chapitre.

L'architecture globale du décodeur est détaillée dans la figure 4.24. Elle est composée d'un unique processeur pour les nœuds de variable connecté à un seul processeur pour les nœuds de parité. L'architecture comporte aussi des modules de mémorisation comprenant d_v mémoires M_{VtoC} pour

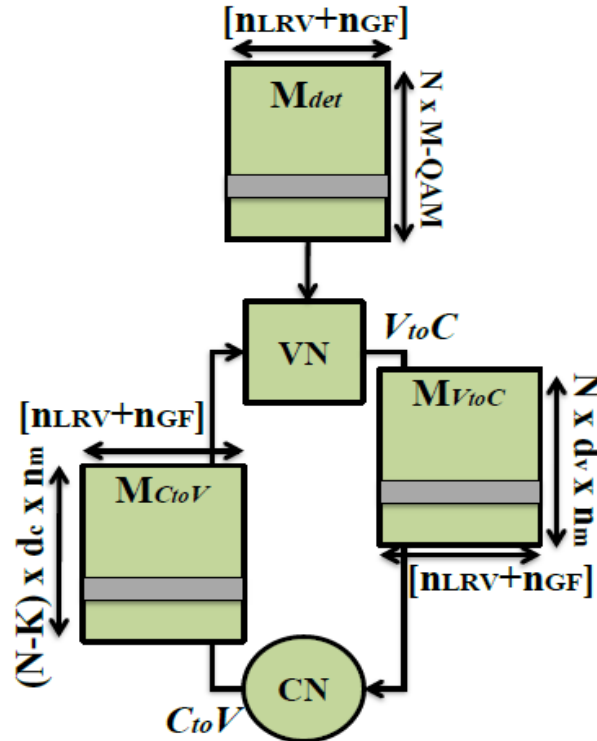


FIGURE 4.24 – Architecture globale du décodeur LDPC non binaire

sauvegarder les messages extrinsèques associés aux nœuds de variable et d_c mémoires M_{CtoV} pour sauvegarder les messages extrinsèques associés aux nœuds de parité. Elle comporte également une autre mémoire M_{det} pour sauvegarder l'information intrinsèque provenant du détecteur MIMO.

Le processus de décodage pour cette architecture comprend deux étapes principales. Au cours d'une étape préliminaire, les messages entrants M_{CtoV} provenant du nœud de parité connecté sont initialisés à zéro, puis :

1-) La première étape consiste à mettre à jour le processeur de nœuds de variable VN , à partir des informations extrinsèques provenant du détecteur MIMO et sauvegardées dans la mémoire M_{det} . De plus, la mise à jour dépend des informations issues du processeur nœud de parité CN obtenues lors de l'itération $t - 1$ et sauvegardées dans les mémoires M_{CtoV} . Les résultats de ce calcul sont sauvegardés dans les mémoires M_{VtoC} .

2-) Durant la deuxième étape, l'information *a posteriori* provenant du nœud de variable est transmise au processeur des nœuds de parité CN . Ainsi, la mise à jour du processeur des nœuds de parité CN à partir des informations provenant du processeur des nœuds de variable VN est effectuée. $d_c - 1$ messages sont mis à jour dans la mémoire M_{CtoV} . Ce processus est répété pour tous les nœuds de parité associés à la représentation du code LDPC non-binaire.

Ainsi, durant chaque inter-itération entre le détecteur et le décodeur, la mise à jour d'un nœud de variable et la mise à jour de deux nœuds de parité sont effectuées. A l'issue de la dernière inter-itération, les valeurs des nœuds de variable déterminent la valeur du mot de code. Nous devons noter que puisque le degré de la connectivité d'un nœud de variable $dv = 2$, le processus de décision ne peut se faire que lors de la mise à jour de la deuxième contribution $CtoV_2$ de chaque nœud de variable.

4.4.2 Système de mémorisation

Chaque symbole du corps de Galois défini sur $GF(64)$ est codé sur $n_{GF} = 6$ bits. En outre, l'information LRV provenant du détecteur relative à un symbole défini sur 64 QAM est également codée sur $n_{LRV} = 6$ bits. Il en découle que la taille de la mémoire M_{det} est égale à $\{(N \times M\text{-QAM}) \times (n_{LRV} + n_{GF})\}$, avec $N = 384$ symboles et $M = 64$.

Rappelons que l'algorithme de décodage *Min-Sum* étendu aussi appelé (EMS) [44] a été appliqué dans cette étude. Cet algorithme offre un bon compromis entre les performances et la complexité calculatoire. En appliquant cet algorithme, chaque message $CtoV$ ou $VtoC$ se compose de $n_{mdec} = 12$ couples (LRV ; GF). Par conséquent, les données associées à un message sont sauvegardées dans n_{mdec} adresses mémoires consécutives. Puisque les informations LRVs sont codés sur $n_{LRV} = 6$ bits et que les indices des symboles sont codés sur $n_{GF} = 6$ bits, la taille de la mémoire M_{CtoV} est $\{(N \times n_{mdec}) \times (n_{LRV} + n_{GF}) \times d_v\}$. De même, la taille de la mémoire M_{VtoC} est $(N - K \times n_{mdec}) \times (n_{LRV} + n_{GF}) \times d_c$, avec $N - K = 192$ correspond au nombre de nœuds de parité pour un rendement $R = 1/2$.

Notons que la troncature des messages entraîne une dégradation des performances qui peut être compensée en utilisant une valeur de fiabilité constante $CtoV.LRV[n_{mdec} - 1] + offset$ pour les symboles non retenus lors de la troncature [82]. $CtoV.LRV[n_{mdec} - 1]$ désigne la valeur LRV de l'information la moins fiable provenant du nœud de parité à l'itération $t - 1$. *offset* est un coefficient déterminé par simulation de façon à obtenir le meilleur WER possible.

4.4.3 Architecture du processeur associé aux nœuds de variable

La figure 4.25 représente l'architecture du processeur associé aux nœuds de variable. Cette architecture se compose de deux registres R1 et R2, un autre registre R3 de 64 valeurs, d'un module de $Tri \{64 \rightarrow 12\}$, d'un second module de $Tri \{12 \rightarrow 12\}$, de deux mémoires *FIFO* de 12 valeurs, d'un comparateur réel, de trois additionneurs réels, d'un multiplexeur et d'un module *décision*. Il est à noter que les architectures de $Tri \{64 \rightarrow 12\}$ et $Tri \{12 \rightarrow 12\}$ sont similaires à celle décrite dans la sous-section 1.3. L'architecture du processeur VN fonctionne selon deux modes :

1-) Mise à jour du processeur VN : Généralement, la mise à jour du processeur VN est faite d'une part par les informations extrinsèques associées au nœud de variable provenant du détecteur MIMO. Ces valeurs sont sauvegardées dans la mémoire M_{det} sous forme de couple $(det.LRV; det.GF)$. D'autre part, la mise à jour dépend des messages $CtoV$ provenant du nœud de parité. L'objectif est de mettre à jour les messages $VtoC$ qui seront sauvegardés dans les mémoires M_{VtoC} .

2-) Décision : Lors de la dernière inter-itération entre le détecteur et le décodeur et à l'aide d'un module *décision*, le processeur VN détermine le symbole décodé après la mise à jour de la dernière contribution $CtoV_2$ pour le nœud de variable.

4.4.3.1 Mise à jour du processeur VN

Lors de la première inter-itération entre le détecteur et le décodeur, le processeur VN reçoit 64 couples $(det.LRV; det.GF)$ du détecteur MIMO. Ces 64 couples sont triés $\{64 \rightarrow 12\}$ afin de générer un message contenant n_{mdec} couples (LRV, GF) selon un ordre croissant. Ce message est envoyé directement au processeur associé aux nœuds de parité CN via un multiplexeur.

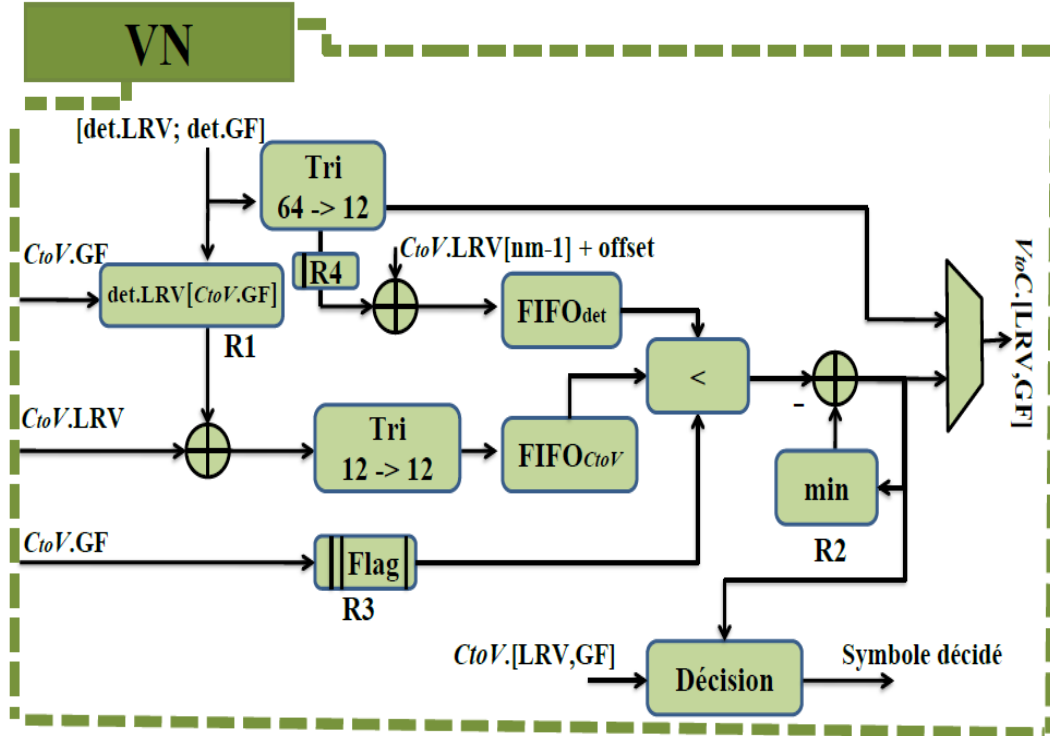


FIGURE 4.25 – Découpe architecturale pour un processeur associé aux nœuds de variable

Dès la deuxième inter-itération, le processeur VN reçoit à nouveau 64 couples $(det.LRV; det.GF)$ du détecteur MIMO. Ces 64 couples sont triés $\{64 \rightarrow 12\}$ afin de générer un message contenant n_{mdec} couples (LRV, GF) selon un ordre croissant. Ce traitement nécessite 64 cycles d'exécution. Le processeur VN doit exécuter successivement deux processus. Le premier consiste à mettre à jour les 12 couples $(det.LRV; det.GF)$ les plus fiables parmi les 64 couples sauvegardées dans la mémoire M_{det} en leur ajoutant la valeur $CtoV.LRV[n_{mdec} - 1] + offset$ afin de compenser la troncature des messages non retenus. Ce traitement nécessite 12 cycles d'exécution. Rappelons que $CtoV.LRV[n_{mdec} - 1]$ est la valeur LRV de l'information la moins fiable provenant du nœud de parité et obtenue lors de l'itération $t - 1$. Le second processus consiste à mettre à jour les LRVs du message $CtoV$ provenant du nœud de parité comme suit :

$$CtoV.LRV[x] = CtoV.LRV[x] + det.LRV[CtoV.GF[x]] \quad (4.2)$$

avec $\{x = 0, 1, ..n_{mdec} - 1\}$. $det.LRV[CtoV.GF[x]]$ correspond aux LRV sauvegardés dans la mémoire M_{det} qui ont des indices GF communs à celles des 12 couples provenant du nœud de parité. Ainsi, le nouveau message $CtoV.LRV[x]$ est réordonné par ordre croissant par le module $Tri \{12 \rightarrow 12\}$. Ce traitement nécessite 12 cycles d'exécution. Ensuite, les deux nouveaux messages $det.LRV$ et $CtoV.LRV$ sont stockés dans deux mémoires $FIFO$ (*First-In First-Out*).

Le module *Flag* est un registre de 64 bits qui sert à indiquer la présence d'un symbole donné dans $CtoV.GF[x]$, $\{x = 0, 1, ..n_{mdec} - 1\}$. En effet, chaque bit du module *Flag* est associé à un symbole GF du corps de Galois. Il y a donc deux cas possibles :

1-) $Flag[det.GF[k]] = 1$, avec $\{k = 0, 1, ..n_{mdec} - 1\}$, alors le symbole est déjà présent dans $CtoV.GF$. Dans ce cas, aucune comparaison n'est faite et $FIFO_{det}$ s'incrmente d'une position pour fournir $det.GF[k + 1]$ au comparateur.

2-) $Flag[det.GF[k]] = 0$, avec $\{k = 0, 1, ..n_{mdec} - 1\}$, alors le symbole n'est pas présent dans $CtoV.GF$. Dans ce cas, il y a deux possibilités :

i-) $(CtoV.LRV[x] < det.LRV[k])$ alors $CtoV.LRV[x]$ est ajoutée à $VtoC$ et la sortie de $FIFO_{CtoV}$ prend la valeur $CtoV.LRV[x + 1]$.

ii-) $(CtoV.LRV[x] > det.LRV[k])$ alors $det.LRV[k]$ est ajoutée à $VtoC$ et la sortie de $FIFO_{det}$ prend la valeur $det.LRV[k + 1]$.

Ce traitement permet d'éviter la présence de redondances des indices GF dans le message $VtoC$. Le module *Compareur* permet, quant à lui, de comparer les sorties des FIFOs et de sélectionner les messages les plus fiables en soustrayant le LRV le plus fiable $\{min\}$ de chaque valeur sortante. Finalement, les 12 nouveaux couples $(VtoC.LRV; VtoC.GF)$ sont envoyés au nœud de parité CN pour un nouveau traitement.

L'organisation temporelle de la mise à jour du processeur VN est présentée dans la figure 4.26. Lors de la mise à jour de la première contribution $CtoV_1$ du nœud de variable, nous avons tout d'abord besoin de trier les informations reçues du détecteur MIMO. Ainsi, le processeur VN commence le *Tri* $\{64 \rightarrow 12\}$ qui dure 64 cycles d'exécution. En parallèle, le processeur VN met à jour les informations $CtoV$ provenant du nœud de parité. Il commence le *Tri* $\{12 \rightarrow 12\}$ dès qu'il reçoit la première valeur mise à jour. Ainsi, 12 cycles d'exécution sont nécessaires pour permettre la sauvegarde des messages triés dans le $FIFO_{CtoV}$.

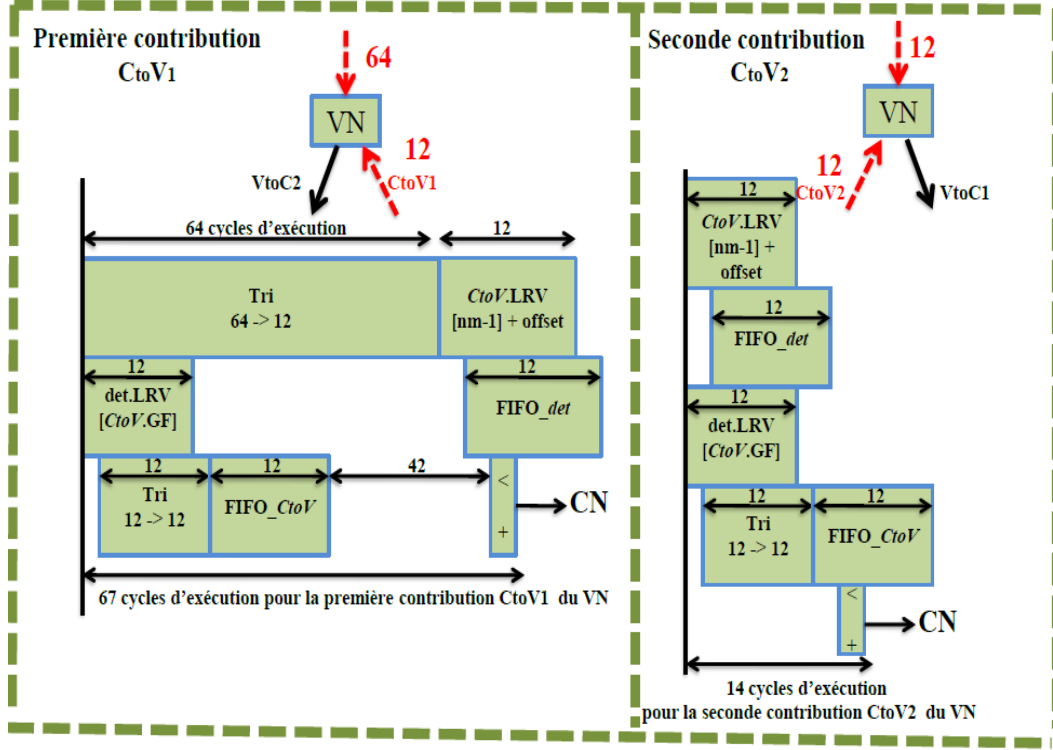


FIGURE 4.26 – Organisation temporelle pour la mise à jour du processeur associé aux nœuds de variable

Lorsque le $Tri \{64 \rightarrow 12\}$ des informations reçues du détecteur MIMO est effectué, la mise à jour des messages débute en leur ajoutant la valeur $CtoV.LRV[n_{mdec} - 1] + offset$. Ce traitement nécessite 12 cycles d'exécution. Dès qu'il reçoit la première valeur mise à jour, le processeur commence la comparaison des valeurs reçues.

En résumé, la mise à jour du processeur VN par la première contribution $CtoV_1$ du nœud de variable nécessite 67 cycles d'exécution.

Lors de la mise à jour de la seconde contribution $CtoV_2$ du nœud de variable, nous bénéficions du $Tri \{64 \rightarrow 12\}$ effectué lors de la mise à jour de la première contribution $CtoV_1$ du nœud de variable. Cela permet un traitement conjoint de plusieurs sous-blocs comme illustré la figure 4.26. Ainsi, 14 cycles d'exécution sont nécessaires pour la mise à jour des couples du message associé à la seconde contribution $CtoV_2$ du nœud de variable.

4.4.3.2 Décision

Le processeur VN a pour but de prendre une décision sur le symbole transmis. Cette décision nécessite la disponibilité des deux messages $CtoV_1$ et $CtoV_2$ lors de la dernière inter-itération. Prenons par exemple le nœud de variable V1 sur lequel le processeur VN doit prendre une décision. Pendant une intra-itération deux messages $CtoV_1$ et $CtoV_2$ associés à V1 sont transmis au processeur VN correspondant. Dès qu'un premier message $CtoV_1$ est disponible, le processeur VN effectue la mise à jour. De plus, il génère un nouveau message correspondant à la somme de la LRV provenant du détecteur et du message $CtoV_1$ que nous avons noté dans l'équation 4.2 $\{CtoV.LRV[x] = CtoV.LRV[x] + det.LRV[CtoV.GF[x]]\}$. Ce message est stocké dans la mémoire M_{VtoC} .

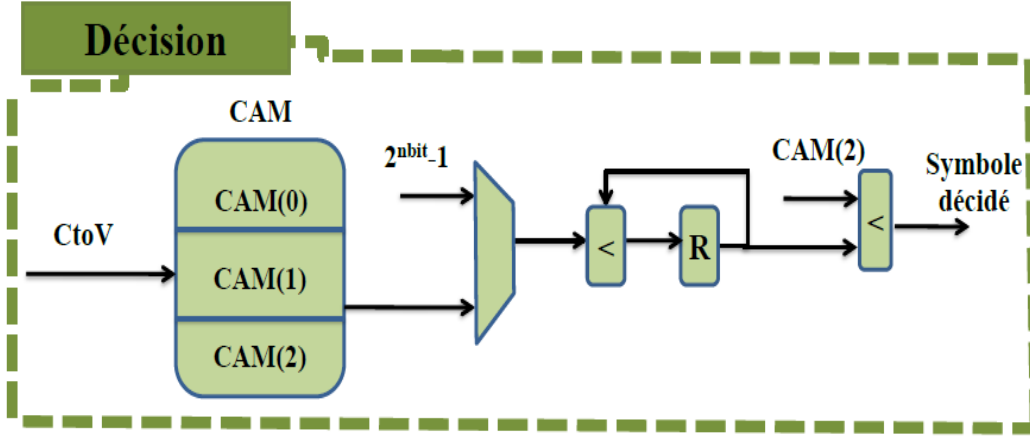


FIGURE 4.27 – Découpe architecturale du bloc décision

Lorsque le processeur CN a effectué son traitement, il produit le message $CtoV_2$ qui sera exploité par le processeur VN pour la décision finale. Une technique traditionnelle de prise de décision consiste à comparer les n_{mdec} couples du message $CtoV_2$ à une mémoire adressable par contenu CAM (*Content Adressable Memory*). Cette mémoire est de taille n_{mdec} et contient la somme du LRV provenant du détecteur et du message $CtoV_1$. L'objectif est d'additionner les LRV si le symbole entrant a trouvé son équivalent dans la mémoire CAM ou dans le cas contraire de saturer les valeurs LRV. Le symbole décidé sera le symbole ayant la valeur LRV la plus faible. Cette méthode nécessite l'utilisation d'une mémoire CAM de taille n_{mdec} , ce qui est très complexe.

Afin de réduire la taille de la mémoire CAM, une méthode de décision sous-optimale a été proposée dans [82]. L'idée est après avoir trié dans l'ordre croissant le message $CtoV_1$, une partie de l'information est suffisante afin de prendre une décision finale. Cette méthode nous permet de réduire la taille de la CAM de n_{mdec} à n_s avec $n_s \ll n_{mdec}$ et de simplifier l'architecture résultante en comparant le message d'entrée avec les $n_s - 1$ premiers symboles GF de la CAM. Dans ce cas, le symbole qui n'a pas trouvé son équivalent dans la CAM est rejeté. A l'issue de n_{mdec} cycles de comparaison, une dernière comparaison est faite entre le minimum sélectionné et la valeur $CAM(n_s - 1).LRV$. Le symbole décidé doit alors avoir la valeur LRV la plus faible. L'architecture proposée dans [82] est présentée dans la figure 4.27.

La valeur *valeur - sat* est la valeur de saturation égale à $\{2^{nbbit} - 1\}$. Cette valeur représente le LRV du symbole le moins fiable. Dans notre étude, nous disposons des indices des corps de Galois. De plus, les informations LRV sont codés sur $nbbit = 6$. Par conséquent, la valeur de saturation du symbole le moins fiable est égale à 63. Cette méthode a été validée par simulation pour différents tailles de CAM [82]. Les simulations analysées dans [82] montrent que $n_s = 3$ est un bon compromis entre performance de correction et complexité. A paritr des ces éléments, nous avons décidé d'appliquer cette méthode dans notre étude avec un CAM de taille $n_s = 3$.

4.4.4 Architecture du processeur associé aux nœuds de parité

4.4.4.1 Etape élémentaire de traitement pour un nœud de parité

L'algorithme EMS peut être implanté d'une manière efficace en utilisant une implantation de type "Forward/Backward", expliqué dans la section 1.4.3.5. Durant chaque étape élémentaire, nous

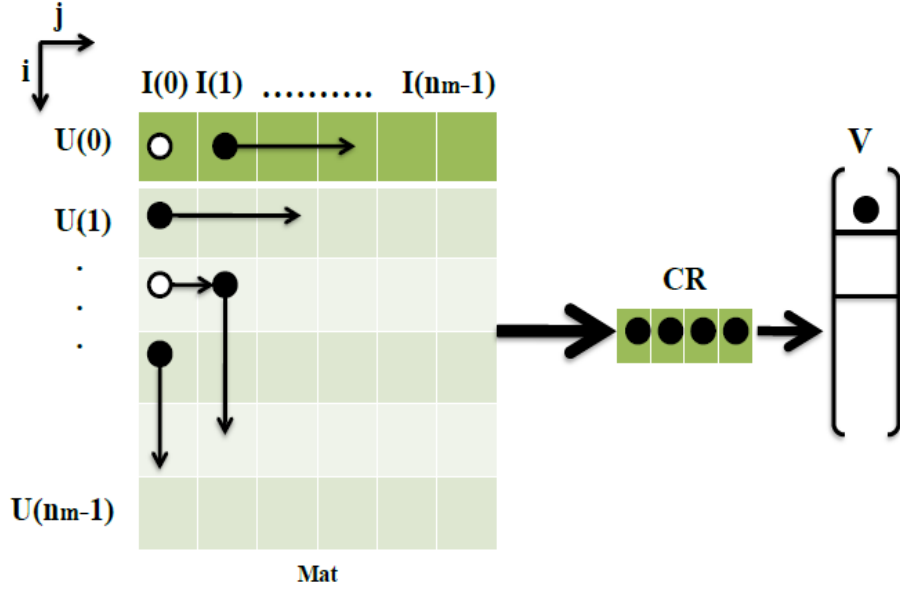


FIGURE 4.28 – Matrice M_{at} associé à l'algorithme *L-Bubble check*

considérons deux messages entrants U et I pour effectuer le calcul de fiabilités. Le principe d'une étape élémentaire est de sélectionner n_{mdec} éléments parmi n_{mdec}^2 candidats. La réalisation classique de cette fonction nécessite un comparateur de taille n_{mdec} exécuté n_{oper} fois, où n_{oper} représente le nombre d'opérations nécessaires à la construction de la liste des n_{mdec} symboles GF , tout en tenant compte de l'élimination de symboles de redondance.

L'algorithme *L-Bubble check* présenté dans [97], [82], montre que la complexité calculatoire peut être réduite de $\{n_{oper} \times n_{mdec}\}$ à $\{n_{oper} \times \psi(n_{mdec})\}$ où $\psi(n_{mdec})$ est le nombre de candidats considérés au cours d'une opération de comparaison avec $\psi(n_{mdec}) = n_{mdec}/2$. Ainsi, $\psi(n_{mdec})$ dépend de la taille de la liste à trier. Un nombre fixe de candidats par exemple 4 est à considérer, indépendamment de la valeur de n_{mdec} . Cet algorithme propose une stratégie de parcours des deux vecteurs triés U et I .

Pour ce faire, nous utilisons une matrice M_{at} construite à partir des vecteurs U et I , telque de M_{at} est de la forme $M_{at}[i, j] = U[i] + I[j]$, avec $\{i = 0, 1, n_{mdec} - 1\}$ et $\{j = 0, 1, n_{mdec} - 1\}$. Cette matrice représente la somme terme à terme de deux vecteurs U et I de taille n_{mdec} . Elle contient les n_{mdec}^2 candidats pour la mise à jour du vecteur sortant V .

Le principe de l'algorithme *L-Bubble check* est d'explorer d'une manière efficace la matrice M_{at} pour calculer itérativement les n_{mdec} valeurs plus fiables de sortie. Cela repose sur le fait que la matrice M_{at} est construite à partir des messages triés. L'étape élémentaire dans [98], [99], repose sur la supposition que les n_{mdec} symboles les plus fiables sont essentiellement distribués sur les deux premières lignes et les deux premières colonnes de la matrice M_{at} . La mise à jour des candidats s'effectue selon 4 parcours de sélection comme le montre la figure 4.28.

La fonction de l'étape primaire basé sur l'algorithme *L-Bubble check* considère un registre de comparaison CR de taille 4 initialisé à $M_{at}(0,0)$, $M_{at}(1,0)$, $M_{at}(2,0)$ et $M_{at}(2,1)$. Durant chaque cycle d'exécution, il effectue une comparaison puis sélectionne l'élément ayant le LRV le plus fiable. L'élément sélectionné est généré en sortie et son voisin est chargé dans le registre de comparaison CR .

Le processus est répété n_{oper} fois. L'algorithme *L-Bubble check* peut être décomposé en trois étapes :

1-) **Initialisation** : tout d'abord, nous initialisons le registre de comparaison $CR(i) = M_{at}(i, 0)$ pour $i = 0, 1, 2$ et $CR(i) = M_{at}(2, 0)$ pour $i = 3$.

2-) **Comparaison** : Le registre de comparaison compare les 4 valeurs sauvegardées et sélectionne l'élément ayant le LRV le plus fiable parmi les LRVs des quatre candidats.

3-) **Chargement** : Remplacement de l'élément extrait $CR(m)$, pour $\{m=0,1,2,3\}$ par son voisin dans la matrice M_{at} selon deux conditions :

i-) Si $(i < 2)$ ou $(i = 2 \text{ et } j = 0)$, le registre CR charge la valeur $CR(M) = M(i, j + 1)$ et $j = j + 1$.

ii-) Si $(i > 2)$ ou $(i = 2 \text{ et } j = 1)$, le registre CR charge la valeur $CR(M) = M(i + 1, j)$ et $i = i + 1$.

Le processus est répété n_{oper} fois afin de remplir le vecteur sortant V de n_{mdec} éléments.

4.4.4.2 Architecture d'une étape élémentaire ECN

Dans cette sous-section, nous allons détailler l'architecture d'une étape élémentaire ECN (*Elementary Check Node*) constituant le processeur des nœuds de parité. Puis, nous décrirons l'architecture du processeur qui comprend plusieurs ECNs.

L'architecture de l'étape élémentaire ECN met en œuvre l'algorithme *L-Bubble check* tel qu'illustré dans la figure 4.29. Cette architecture se compose de deux blocs de mémoire M_U et M_I pour sauvegarder les n_{mdec} couples (LRV, GF) constituant l'entrée des messages U et I . La table d'indices

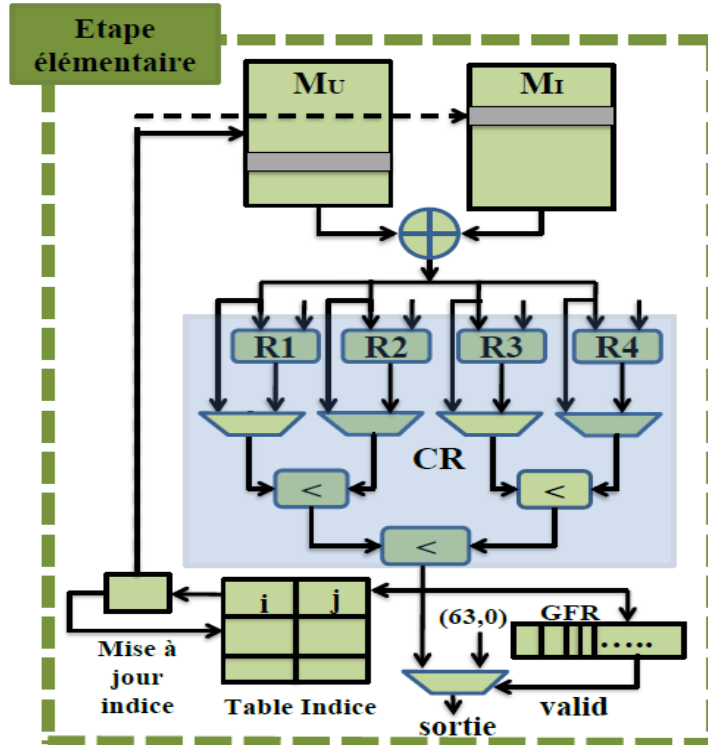


FIGURE 4.29 – Architecture d'une étape élémentaire du processeur ECN associé aux nœuds de parité

sert à stocker les indices du contenu du registre de comparaison CR en fonction de leur position dans la matrice M_{at} . Ces indices serviront à générer les adresses de lecture des blocs de mémoires M_U et M_I .

Comme nous l'avons déjà expliqué précédemment, le registre CR est initialisé selon l'étape 1 de l'algorithme *L-Bubble check* en utilisant un additionneur pour sommer les données provenant des mémoires M_U et M_I . Il est à noter que le registre CR stocke la valeur *LRV* avec son indice *GF* associé à chaque symbole. Le registre de comparaison compare les 4 valeurs sauvegardées et sélectionne l'élément ayant le LRV le plus fiable parmi les LRV des quatre candidats. Ensuite, la table d'indice est mise à jour en fonction de la position du couple sélectionné dans le registre CR, comme décrit précédemment à l'étape 3. Ce processus continue jusqu'à la génération d'une liste de n_{mdec} éléments.

Un registre à 64 valeurs GFR est mis en œuvre pour éviter la perte de symboles de redondance. Dès la génération d'un symbole à GF, son *flag* associé à la GFR est mis à 1. Puis, si le même symbole est ré-généré, le signal *valid* indique la présence d'un symbole de redondance. Dans ce cas, un symbole non-valide ayant comme $LRV = 63$ et $GF = 0$ est transmis. Afin de compenser l'effet de redondance, certaines opérations supplémentaires peuvent être effectuées pour générer n_{mdec} symbole GF validé. En résumé, n_{oper} cycles d'exécution sont nécessaires afin de générer n_{mdec} symboles GF validés, avec $n_{oper} = n_{mdec} + S_{oper}$ où $S_{oper} = 4 > 0$.

4.4.4.3 Architecture d'un processeur associé aux nœuds de parité CN

L'architecture d'un processeur de nœud de parité de degré de connectivité $d_c = 4$ nécessite la mise en œuvre de $\{3 \times (d_c - 2)\}$ processeurs d'une étape élémentaire ECN regroupés dans $(d_c - 2)$ blocs. Ainsi, trois processeurs ECN sont associés à un bloc. La figure 4.30 présente l'architecture du processeur CN de degré de connectivité 4.

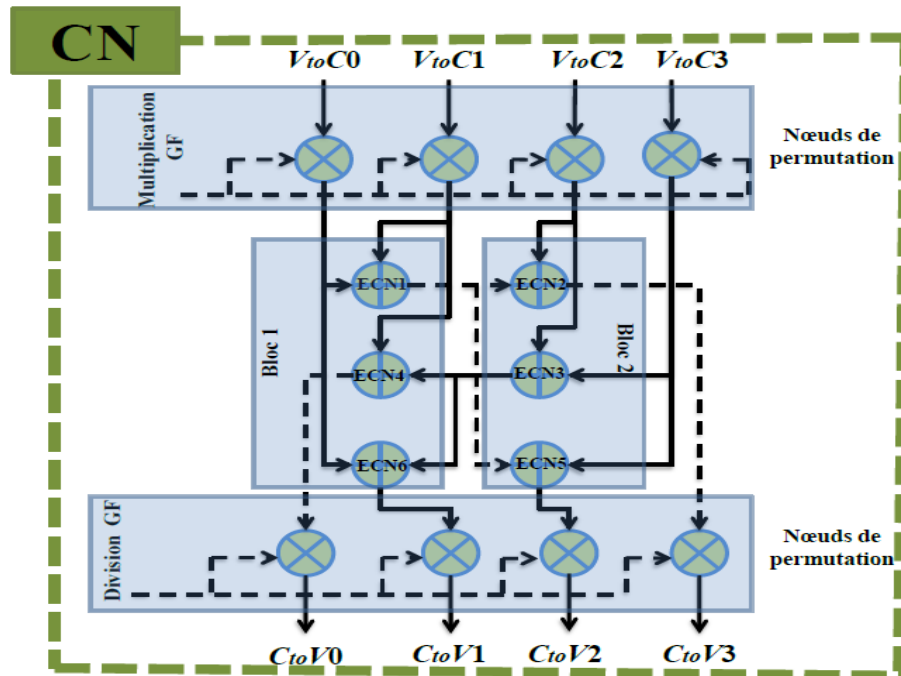


FIGURE 4.30 – Architecture d'un processeur associé aux nœuds de parité

Les étapes élémentaires ECN dans chaque bloc sont organisées selon trois couches. Les messages $VtoC_i$, $\{i = 0, \dots, 3\}$ sont fournis en parallèle au processeur CN. Ainsi, les symboles de GF sont multipliés par les coefficients de la matrice de contrôle de parité (nœud de permutation) en utilisant 4 multiplicateurs GF. A noter que l'architecture retenue pour le multiplieur GF est celle décrite dans [97], [82]. Les deux blocs reçoivent les messages et chaque bloc commence son calcul dès que les deux messages d'entrée sont disponibles.

Prenons par exemple l'exécution de la première étape élémentaire ECN1 du Bloc1. Cette ECN reçoit simultanément les messages $VtoC_0$ et $VtoC_1$. Elle commence immédiatement le calcul. Après 2 cycles d'exécution, ECN1 génère le premier couple (LLR, GF) qui est envoyé à la ECN2 du Bloc2. La ECN2 du Bloc2 sera exécutée de manière similaire à partir de l'information provenant du bloc1 et du message $VtoC_2$. Enfin, le message $CtoV_3$ est généré après une durée de latence de $2 \times (d_c - 2) = 4$ cycles d'exécution. Cette étape est nommée *Forward*, comme expliqué dans le chapitre 1.

Parallèlement, l'étape *Backward* débute avec le calcul de la ECN3 du deuxième bloc. Le message $CtoV_0$ est généré après 4 cycles d'exécution. En ce qui concerne le calcul des ECN5 et ECN6, il débute 2 cycles plus tard. Après avoir fait les divisions des symboles GF par les coefficients de la matrice de contrôle de parité, les messages $CtoV_i$, $\{i = 0, \dots, 3\}$ sont envoyés parallèlement au processeur VN. Notons que l'architecture retenue dans cette étude pour les 4 diviseurs GF est celle qui a été décrite dans [97].

L'organisation temporelle de la mise à jour du processeur CN est donnée dans la figure 4.31. Les messages $VtoC_i$, $\{i = 0, \dots, 3\}$ sont transmis parallèlement au processeur CN. Deux cycles d'exécution sont nécessaires pour la multiplication et pour la division des symboles GF par les coefficients de la matrice de contrôle de parité. Puis, l'exécution des étapes élémentaires ECN1 et ECN2 nécessite 4 cycles d'exécution. Les étapes élémentaires ECN3 et ECN4 sont exécutées en parallèle. Les étapes élémentaires ECN5 et ECN6 s'exécutent dès qu'elles reçoivent leurs premiers couples disponibles après 2 cycles d'exécution supplémentaires. Au final, 7 cycles d'exécution sont nécessaires pour générer le

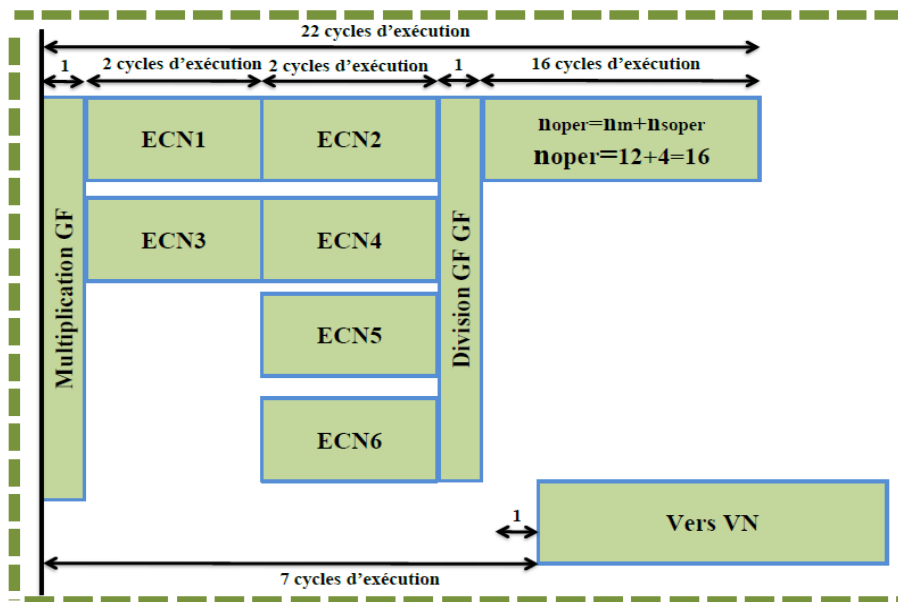


FIGURE 4.31 – Organisation temporelle de la mise à jour d'un processeur associé aux nœuds de parité

premier couple (LRV, GF) . En outre, n_{oper} cycles d'exécution sont nécessaires afin de générer n_{mdec} symboles GF validés, avec $n_{oper} = n_{mdec} + S_{oper}$ où $S_{oper} = 4$. Par conséquent, pour générer la liste composée de n_{mdec} couples (LRV, GF) , 22 cycles d'exécution sont nécessaires au sein du processeur CN.

4.5 Conclusion

Dans ce chapitre, nous avons proposé des solutions architecturales pour les différents modules constituant le récepteur itératif. Tout d'abord, nous avons présenté les différents blocs de haut niveau constituant le récepteur en donnant une vue globale de l'architecture retenue. Cette architecture se compose d'une première partie qui correspond à l'architecture du détecteur MIMO et d'une seconde pour l'architecture de décodage d'un code LDPC-NB.

Puis, nous avons décrit les différents blocs de l'architecture du détecteur MIMO qui correspond à une détection simplifiée basée sur les techniques CI et REI (CI-REI). A l'aide de cette architecture, nous pouvons calculer le rapport de vraisemblance du point initial d'une constellation en considérant la méthode de sous-régions dynamiques. Ainsi, tous les points voisins du point détecté peuvent être obtenus par un calcul récursif. Une représentation architecturale, une organisation temporelle et l'utilisation des ressources arithmétiques et de mémorisation ont été fournies pour chaque bloc constituant le détecteur MIMO.

Enfin, nous avons détaillé les différents blocs de l'architecture du décodeur LDPC non-binaire en considérant un seul processeur pour les nœuds de variable et un seul processeur pour les nœuds de parité. Cette architecture s'inspire fortement de celle réalisée dans le cadre du projet Européenne Davinci. Puis, une étude de l'organisation des échanges d'information entre le détecteur et le décodeur a été faite. Cette étude permet de proposer une architecture globale pour le récepteur itératif avec un échange efficace d'information par le biais d'un parallélisme favorisant la montée en débit et la maîtrise de la latence du récepteur.

Le tableau 4.1 récapitule la latence et la complexité matérielle de chaque bloc du récepteur itératif. Cette synthèse illustre le nombre de cycles d'exécution et les ressources arithmétiques

	Latence cycles d'exécution	Complexité (ressources matérielles)								
		Add	Mult		Div.GF	Comp.	Trans.	Reg.		FIFO
			R	GF				1	64	
Bloc CID (1, 2, 3)	139	3	3			12	1	18	2	
Bloc CID (4)	150	2	1					1		
Bloc AT	92	1				1		2	1	
Bloc AR	64	1							2	
Détecteur MIMO-BP	139 + 150 + 92 × (iter-1)	7	4	0	0	13	1	21	5	0
VN	67+14	1				25		25	2	2
CN	22	6		4	4	8		12	6	
Décodeur LDPC-NB	125 × (iter)	7	0	4	4	33	0	37	8	2

TABLE 4.1 – Synthèse des caractéristiques des différents blocs pour le récepteur itératif

nécessaires pour chaque bloc. L'architecture du détecteur MIMO se compose au total de sept additionneurs réels, de quatre multiplieurs réels, de vingt et un registres à une valeur, de cinq registres à 64 valeurs, d'un transcodeur et de treize comparateurs. De plus, l'architecture du décodeur LDPC-NB se compose de sept additionneurs réels, de quatre multiplieurs GF, de quatre diviseurs GF, de deux FIFO, de trente sept registres à une valeur, de huit registres à 64 valeurs et de trente trois comparateurs.

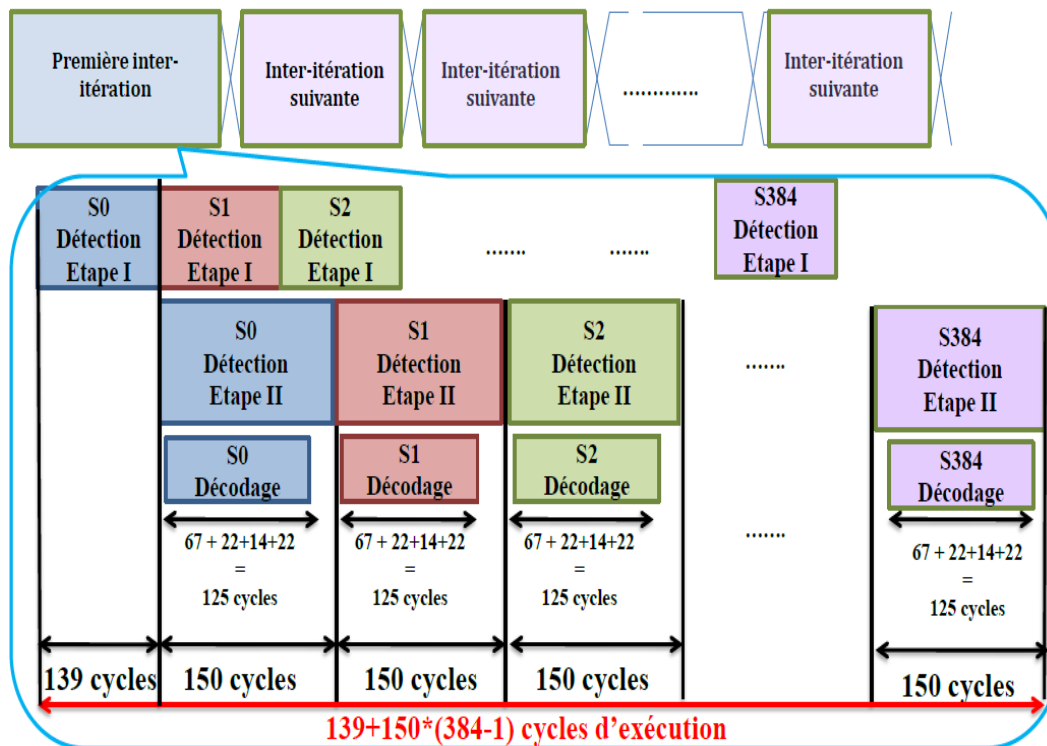


FIGURE 4.32 – Organisation temporelle de l’ordonnancement itératif pour la première inter-itération du récepteur MIMO-BP

Par ailleurs, la figure 4.32 présente l'ordonnancement itératif de la première inter-itération du récepteur MIMO-BP. Tout d'abord, les étapes 1, 2, et 3 du bloc CID, correspondant à l'étape 1 de la détection, nécessitent 139 cycles d'exécution à l'issue desquels les vecteurs LRV envoyés au décodeur sont disponibles. Après ces 139 cycles d'exécution, le décodeur commence à décoder le premier symbole candidat qui nécessite 125 cycles d'exécution. En parallèle, le bloc 4 s'exécute pour calculer par récursion les points voisins des distances euclidiennes. Cela nécessite 150 cycles d'exécution. De manière similaire, nous traitons tous les symboles de la trame courante en détectant et décodant de manière conjointe. Ainsi, la durée totale de la première inter-itération est de $\{(139 + 150 \times 2(\text{couple})) \times 192(\text{symboles})\}$ cycles d'exécution.

Ensuite, pour les autres inter-itérations distinctes de la première inter-itération, nous mettons uniquement à jour l'informations sauvegardées dans le bloc Mémoire. Ce traitement nécessite 92 cycles d'exécution à l'issue desquels les 64 valeurs LRV devant être envoyées au décodeur sont disponibles. En revanche, nous n'avons plus besoin d'attendre la génération des 64 valeurs du détecteur MIMO. Dès que nous recevons la première valeur LRV générée par le détecteur, le décodeur commence à décoder le symbole candidat. L'étape de mise à jour de chaque symbole nécessite 154 cycles d'exécution comme illustré dans la figure 4.33. Ainsi, la durée totale pour la mise à jour des autres

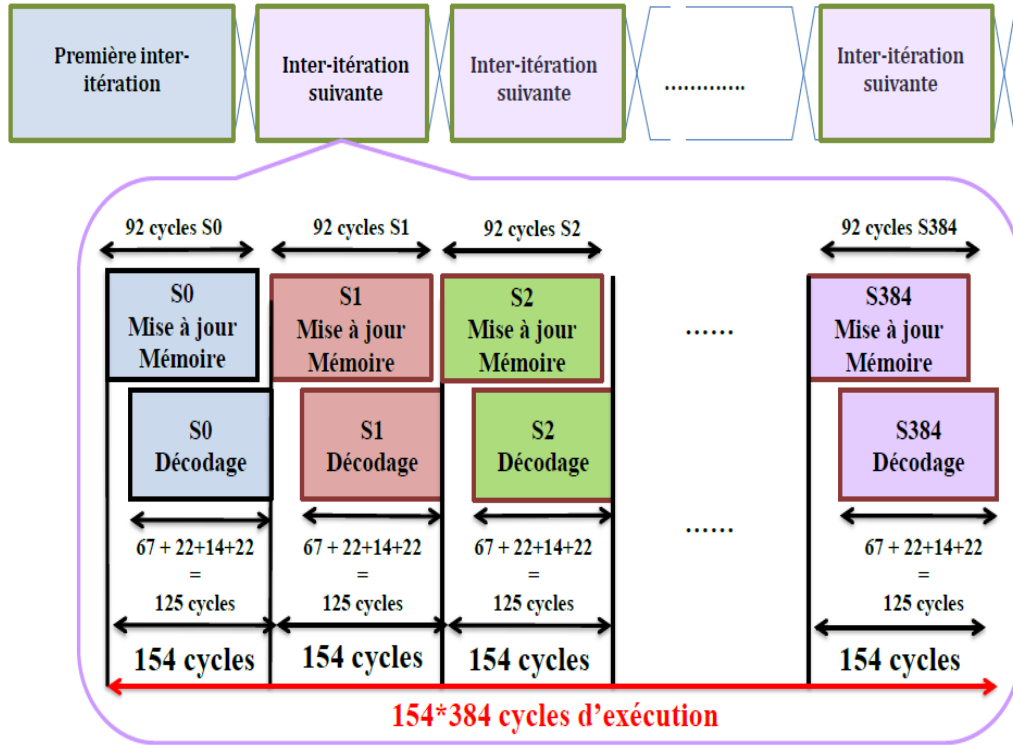


FIGURE 4.33 – Organisation temporelle de l’ordonnancement itératif pour les autres inter-itérations du récepteur MIMO-BP

inter-itérations, distinctes de la première inter-itération, est de $\{(154 \times 2(\text{couple})) \times 192(\text{symboles})\}$ cycles d’exécution. De plus, une analyse de la complexité au niveau des ressources arithmétiques a montré une division par dix du nombre total des ressources arithmétiques par rapport au détecteur de référence MMSE [100] comme illustré dans le tableau 4.2.

	Complexité (ressources matérielles)						
	Add.R	Mult.R	Div.R	Comp.	Trans.	Reg.	
						1	64
Détecteur MIMO-BP	7	4	0	13	1	21	5
Détecteur MMSE	90	52	2	0	0	40	0

TABLE 4.2 – Comparaison des caractéristiques des différents détecteurs pour le récepteur itératif

Les travaux restants concernent l’implémentation matérielle de l’architecture proposée pour le récepteur itératif en précisant le nombre de bits nécessaire pour les informations intrinsèques au sein du détecteur MIMO ainsi que pour les échanges d’information entre le détecteur et le décodeur. Ce travail complémentaire est indispensable pour la valorisation scientifique de notre découpe architecturale.

Conclusion et Perspectives

Cette thèse porte sur l'étude d'algorithmes de réception itératifs dans le contexte d'une transmission MIMO incluant un code correcteur d'erreurs de type LDPC non-binaire. Le travail a été décomposé en deux étapes : une étude algorithmique et une étude des solutions architecturales. Une forte interaction entre ces étapes a été nécessaire pour permettre d'une part une reformulation algorithmique et d'autre part la proposition de solutions architecturales efficaces.

Dans la première partie du manuscrit, nous avons étudié la combinaison d'un détecteur MIMO basé sur la détection symbole par symbole avec un décodeur de code LDPC-NB. Cette étude a été effectuée pour différents algorithmes de détection et selon différents séquencements au niveau du récepteur itératif. Dans ce contexte, nous avons considéré comme référence une détection basée sur l'algorithme de maximum de vraisemblance. Nous avons montré dans un premier temps que le traitement symbole par symbole améliore les performances du système non-itératif par rapport à un traitement bit par bit. En outre, la pénalité introduite par une détection sous-optimale basée sur l'algorithme de propagation de croyance peut atteindre 1 dB dans certains cas. Cependant, elle est fortement réduite à moins de 0.3 dB lorsqu'un traitement itératif est appliquée entre le détecteur et le décodeur.

Dans un second temps, nous avons comparé les performances et la complexité calculatoire de récepteurs MIMO itératifs reposant sur deux séquencements différents pour l'échange des messages entre le détecteur et le décodeur : séquencement "*flooding*" et séquencement "*shuffle*". Nous avons montré lors de la comparaison que le séquencement *shuffle* est plus pertinent au niveau des intra-itérations et des inter-itérations au sein du récepteur MIMO itératif. Pour ce faire, nous avons exploité une représentation de type graphe à facteur conjoint du récepteur MIMO avec un multiplexage spatial suivi d'un décodage de code LDPC-NB. Le principal intérêt de ce séquencement est de favoriser la convergence en divisant par trois la latence d'exécution pour des performances équivalentes. Cette étude effectuée pour différents algorithmes de détection et selon différents séquencements au niveau du récepteur itératif a fait l'objet de plusieurs publications [75], [76].

Dans une deuxième partie du manuscrit, nous avons proposé plusieurs algorithmes visant à réduire la complexité calculatoire du récepteur itératif. Ces algorithmes reposent sur deux principaux niveaux. Le premier niveau est le calcul initial des distances euclidiennes. Le second niveau est celui des échanges d'information entre le détecteur et le décodeur au cours de chaque inter-itération. Nous avons ainsi proposé plusieurs méthodes visant à réduire la complexité du calcul initial des distances euclidiennes pour une constellation d'ordre élevé (64 QAM) à deux dimensions. Une comparaison des performances et de la complexité calculatoire pour les différentes méthodes proposées a été effectuée. Puis, une technique visant à réduire le nombre d'échanges d'information entre le détecteur et le décodeur a été proposée. A l'aide de diagrammes EXIT, un jeu de paramètres pour l'échange d'informations au sein du récepteur itératif a été retenu. Une analyse de la complexité calculatoire a montré une division par 20 du nombre total d'opérations par itération au sein du récepteur itératif par rapport à l'algorithme de référence basé sur la propagation de croyance. Ces résultats ont fait

l'objet de plusieurs publications [81], [78], [80].

La suite de l'étude s'est attachée à proposer des solutions architecturales pour les différents modules constituant le récepteur itératif retenu. Deux niveaux ont été définis dans cette étude. Tout d'abord, nous avons présenté l'architecture des différents blocs associés au détecteur MIMO. A l'aide de cette architecture, il est possible d'effectuer le calcul du rapport de vraisemblance du point initial d'une constellation en considérant une méthode de sous-régions dynamiques. Dès lors, tous les points voisins du point détecté peuvent être déduits à l'aide d'un calcul récursif. Puis, nous avons détaillé les différents blocs de l'architecture du décodeur LDPC non-binaire. Un seul processeur a été alloué pour les nœuds de variable et un processeur permet de traiter tous les nœuds de parité. Cette architecture s'inspire fortement de celle développée dans le cadre du projet Européen Davinci. Au final, une architecture globale a été définie pour le récepteur MIMO itératif. Cette dernière permet d'exploiter le parallélisme de symbole afin de mutualiser les informations extrinsèques et les ressources de calcul comme expliqué dans le chapitre 4. Une analyse de la complexité au niveau des ressources arithmétiques a montré une division par dix du nombre total des ressources arithmétiques par rapport au détecteur de référence MMSE [100]. Ce choix de conception favorise la montée en débit et la maîtrise de la latence du récepteur itératif.

Travaux futurs

La prochaine étape concerne l'implémentation matérielle de l'architecture proposée pour le récepteur itératif. Avant cela, une étude du raffinement de la représentation en virgule flottante vers une représentation en virgule fixe sera nécessaire. Ces travaux supplémentaires sont primordiaux pour la valorisation scientifique de notre découpe architecturale du récepteur itératif.

Sur la base des résultats obtenus lors de l'implémentation matérielle, une nouvelle tâche consistera en l'optimisation de l'architecture au niveau des ressources matérielles et/ou de la montée en débit. Ce travail nécessitera éventuellement de revoir les choix pour les différents paramètres afin d'aboutir au meilleur compromis entre performance de détection-correction, débit et consommation d'énergie.

Par ailleurs, les résultats dans cette étude ont été obtenus en considérant uniquement le cas de deux antennes à l'émission ($N_t = 2$) et de deux antennes à la réception ($N_r = 2$) avec une modulation 64 *QAM* et un corps de Galois d'ordre ($GF = 64$). Or, nous savons que la complexité des algorithmes de détection est fonction de l'ordre de la modulation et du nombre d'antennes à l'émission (N_t). De plus, la complexité de l'algorithme de décodage est fonction de l'ordre des corps de Galois (GF). C'est pourquoi nous proposons de poursuivre nos travaux en considérant un nombre d'antennes plus élevés, des trames de différentes tailles et différents types de modulations basées sur d'autres ordres de corps de Galois.

Liste des publications

Communication dans une conférence internationale avec comité de lecture

1. A. Haroun, C. Abdel Nour, M. Arzel and C Jogo, "Symbol-based BP Detection for MIMO Systems associated with Non-Binary LDPC Codes", in IEEE Wireless Communications and Networking Conference, WCNC2014, Avril 2014, Turquie.
2. A. Haroun, C. Abdel Nour, M. Arzel and C Jogo, "Low-Complexity layered BP-based Detection and Decoding for a NB-LDPC Coded MIMO System", in IEEE Internaional Conference on Communications, ICC2014, Juin 2014, Australie.
3. A. Haroun, C. Abdel Nour, M. Arzel and C Jogo, "Low-Complexity LDPC-coded Iterative MIMO Receiver Based on Belief Propagation algorithm for Detection", accepté à IEEE 8th International Symposium on Turbo Codes and Iterative Information Processing, ISTC2014, Aout 2014, Allemagne.

Communication dans une conférence nationale avec comité de lecture

1. A. Haroun, C. Abdel Nour, M. Arzel and C Jogo, "Réception itérative MIMO basée sur la propagation de croyance et des codes LDPC non-binaires", dans 24ème Colloque Grets, Grets2013, Septembre 2013, France.
2. A. Haroun, C. Abdel Nour, M. Arzel and C Jogo, "An efficient MIMO receiver based on BP algorithm with truncated message-passing", dans 9ème Colloque GDR SoC-SiP, juin 2014, Paris.

Bibliographie

- [1] E. T. AR and I. E. Telatar, "Capacity of multi-antenna gaussian channels," *European Transactions on Telecommunications*, vol. 10, pp. 585–595, 1999. (Cité en pages [xii](#), [18](#) et [20](#).)
- [2] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 2, pp. 311–335, 1998. (Cité en pages [xii](#), [18](#), [19](#) et [20](#).)
- [3] C. Berrou, A. Glavieux, and P. Thitimajshima., "Near shannon limit error-correcting coding and decoding : Turbo-codes." *IEEE International Conference on Communications*, vol. 2, pp. 1064–1070, 1993. (Cité en pages [xii](#), [6](#) et [29](#).)
- [4] C. Douillard, M. Jezequel, C. Berrou, P. Didier, and A. Picart, "Iterative correction of intersymbol interference : turbo-equalization," *European Transactions on Telecommunications*, vol. 6, pp. 507–512, 1995. (Cité en pages [xii](#) et [29](#).)
- [5] G. Foschini and G. Rayleigh, "Layered space-time architecture for wireless communication in a fading environment when using multielement antennas," *Bell Labs Tech. J.*, vol. 1, pp. 41–59, 1996. (Cité en pages [xii](#), [18](#), [20](#) et [21](#).)
- [6] B. Steingrimsson, Z.-Q. Luo, and K. Wong, "Soft quasi-maximumlikelihood detection for multiple-antenna wireless channels," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2710–2719, 2003. (Cité en pages [xii](#), [24](#) et [81](#).)
- [7] S. Gounai and T. Ohtsuki, "Convergence acceleration of iterative signal detection for mimo system with belief propagation," *GLOBECOM*, pp. 1–5, 2006. (Cité en pages [xii](#) et [81](#).)
- [8] R. Gallager, "Low density parity check codes," *Transaction on Information Theory*, vol. 8, pp. 21–28, 1962. (Cité en pages [xii](#), [6](#), [9](#) et [14](#).)
- [9] M. Davey and D. MacKay, "Low density parity check codes over $gf(q)$," *IEEE Communications Letters*, vol. 2, pp. 165–167, 1998. (Cité en pages [xii](#), [11](#), [14](#), [16](#) et [38](#).)
- [10] D. MacKay and M. Davey, "Evaluation of gallager codes for short block length and high rate applications," *Syst. and Graph. Models In Codes*, pp. 113–130, 1999. (Cité en pages [xii](#) et [11](#).)
- [11] Z. Guo and P. NilssonGuo, "A vlsi architecture of the schnorr-euchner decoder for mimo systems," *IEEE Circuits and Systems Symposium on Emerging Technologies : Frontiers of Mobile and Wireless Communication*, vol. 1, pp. 65–68, 2004. (Cité en pages [xii](#), [81](#) et [82](#).)
- [12] J. R. Partington, "Linear operators and linear systems," *London Mathematical Society Student Texts Cambridge University Press*, vol. 75, pp. 521–546, 2004. (Cité en page [1](#).)
- [13] L. Boithias, "Radio wave propagation," McGraw-Hill, 1987. (Cité en page [1](#).)
- [14] P. Guguen, "Techniques multi-antennes émission - réception - applications aux réseaux domestiques sans fil," Ph.D. dissertation, Institut National des Sciences appliquées de Rennes, 2003. (Cité en page [1](#).)
- [15] J. Rayleigh, "For his investigations of the densities of the most important gases and for his discovery of argon in connection with these studies," The Nobel Prize in Physics 1904. (Cité en page [1](#).)
- [16] B. L. Floch, M. Alard, and C. Berrou, "Coded orthogonal frequency division multiplex," *IEEE proceedings*, vol. 83, pp. 982–996, 1995. (Cité en page [2](#).)
- [17] O. Berder, C. Boudier, and G. Burel, "Identification of frequency hopping communications," *WSEAS Conference on circuits systems, communications and computers*, pp. 3851–3856, 2000. (Cité en page [2](#).)

-
- [18] M. Badino, “The foundational role of ergodic theory,” *Foundations of science*, 2005. (Cité en page 2.)
- [19] V. Erceg, L. Greenstein, S. Tjandra, S. Parkoff, A. Gupta, B. Kulic, A. Julius, and R. Bianchi, “An empirically based path loss model for wireless channels in suburban environment,” *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1205–1211, 1999. (Cité en page 2.)
- [20] H. Bölcskei and A. Paulraj, “The communications handbook.” CRC Press, 2002. (Cité en page 2.)
- [21] J. B. Anderson and S. M. Hladick, “Tailbiting map decoders,” *IEEE Journal on selected Area in Communications*, vol. 16, pp. 297–302, 1998. (Cité en page 3.)
- [22] T. V. Kevin McClaning, “Radio receiver design,” Noble Pub. Corp., 2000. (Cité en page 4.)
- [23] E. Viterbo and J. Boutros, “Signal space diversity : a power and bandwidth efficient diversity technique for the rayleigh fading channel,” *IEEE Journal on information Theory*, vol. 44, pp. 1453–1467, 1998. (Cité en page 4.)
- [24] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948. (Cité en pages 5 et 19.)
- [25] R. Hamming, “Error-detecting and error-correcting codes,” *Bell System Technical Journal*, vol. 2, pp. 147–160, 1950. (Cité en page 5.)
- [26] M. Demazure, “Cours d’algebre,” *Cassini*, 1997. (Cité en page 6.)
- [27] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *J. Soc. Ind. Appl. Math.*, vol. 8, pp. 300–304, 1960. (Cité en page 6.)
- [28] A. J. Viterbi, “Error bounds for convolutional codes and asymptotically optimum decoding algorithm.” *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, 1967. (Cité en pages 6 et 24.)
- [29] C. Douillard, M. Jezequel, C. Berrou, N. Brengarth, J. Tusch, and N. Phamnéuds, “The turbo code standard for dvb-rs,” *International Symposium on Turbo codes and Related Topics*, pp. 535–538, 2000. (Cité en page 6.)
- [30] Mackay and Neal, “Near shannon limit performance of low density parity-check codes.” *IEEE Electronics letters*, vol. 32, pp. 1945–1946, 1996. (Cité en pages 6 et 11.)
- [31] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” *IEEE Trans. on Information Theory*, vol. 42, pp. 429–445, 1996. (Cité en page 6.)
- [32] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Transactions on Information Theory*, vol. 27, pp. 533–547, 1981. (Cité en pages 7, 8, 9 et 12.)
- [33] A. Anastasopoulos, “A comparison between the sum-product and the min-sum iterative detection algorithms based on density evolution,” *IEEE Global Telecommunications Conference*, pp. 1021–1025, 2001. (Cité en page 10.)
- [34] N. Wiberg, “Codes and decoding on general graphs.” PhD thesis, Linköping University, Sweden, 1996. (Cité en page 10.)
- [35] R. Zarubica, R. Hinton, S. Wilson, and E. Hall, “Efficient quantization schemes for ldpc decoders,” *IEEE Military Communications Conference*, vol. 16, pp. 1–5, 2008. (Cité en page 10.)
- [36] R. Gallager, “Low density parity check codes,” MIT Press 1963. (Cité en page 10.)
- [37] E. Galois, “Ecrits et mémoires mathématiques d’évariste galois,” Gauthier-Villars, Paris, 1962. (Cité en pages 11 et 12.)
- [38] X.-Y. Hu and E. Eleftheriou, “Binary representation of cycle tanner-graph gf (2q) codes,” *IEEE International Conference on Communications*, 2004. (Cité en page 11.)

-
- [39] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular (2,dc)- ldpc codes over $gf(q)$ using their binary images," *IEEE Transaction on Communication*, vol. 53, pp. 209–213, 2005. (Cité en page 11.)
 - [40] L. Hanzo, T. H. Liew, B. L. Yeap, and S. X. Ng, "Turbo coding, turbo equalisation and space-time coding for transmission over fading channels," *Wiley & IEEE*, vol. 9, pp. 317–390, 2002. (Cité en pages 12 et 38.)
 - [41] D. Sridhara and T. E. Fuja, "Low density parity check codes over groups and rings," *Information Theory Workshop*, 2002. (Cité en page 12.)
 - [42] E. Boutillon, J. Castura, and F. R. Kschischang, "Decoder-first code design," *IEEE International Symposium on Turbo Codes and Related Topics*, pp. 453–462, 2000. (Cité en page 12.)
 - [43] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X. Y. Hu, "Reduced complexity decoding of ldpc codes," *IEEE Transactions on Communications*, vol. 53, pp. 1288–1299, 2005. (Cité en page 12.)
 - [44] D. Declercq and M. Fossorier, "Decoding algorithms for non-binary ldpc codes over gf ," *IEEE Transaction on Communication*, vol. 55, pp. 633–643, 2007. (Cité en pages 12, 14, 39 et 103.)
 - [45] A. Voicila and D. Declercq, "Benchmark non-binary ldpc codes : Report and software," *DaVinci Deliverable D4.1, Europ. project FP7*, 2008. (Cité en pages 12, 14, 30 et 102.)
 - [46] H. Anton and C. Rorres., "Elementary linear algebra," *Applications Version. John Wiley & Sons*, 2000. (Cité en page 13.)
 - [47] C. Berrou, "Codes et turbo codes," *1e édition, Springer-Verlag*, 2007. (Cité en page 13.)
 - [48] D. Declercq and M. Fossorier, "Extended minsum algorithm for decoding ldpc codes over $gf(q)$," *IEEE International Symposium on Information Theory*, 2005. (Cité en pages 15 et 16.)
 - [49] A. Voicila, D. Declercq, M. Fossorier, and F. Verdier, "Décodage des codes ldpc non binaires : un algorithme à très faible complexité," GRETSI, Troyes, France, 2007. (Cité en pages 15, 37 et 71.)
 - [50] J. H. WINTERS, "Optimum combining in digital mobile radio with cochannel interference," *IEEE Trans. on Vehicular Technology*, 198. (Cité en page 17.)
 - [51] L. Zheng and C. Tse, "On the capacity of radio communication systems with diversity in a rayleigh fading environment," *IEEE GLOBECOM*, 1987. (Cité en page 21.)
 - [52] G. J. Focshini, G. D. Golden, R. Valenzuela, and Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE Journal on Selected Areas on Communications*, vol. 17, pp. 1841–1852, 1999. (Cité en page 21.)
 - [53] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1451–1458, 1998. (Cité en pages 22 et 23.)
 - [54] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space time codes for high data rate wireless communication : performance criterion and code construction," *IEEE Transactions on Information Theory*, vol. 44, pp. 744–765, 1998. (Cité en page 22.)
 - [55] S. Kaiser, "Ofdm code division multiplexing in fading channels," *IEEE Transactions on communications*, 2002. (Cité en page 23.)
 - [56] V. L. NIR, "Etude et optimisation des systèmes multi-antennes associés à des modulation multiporteuses," *Thèse de doctorat, INSA de Rennes*, 2004. (Cité en page 23.)
 - [57] H. L. V. Trees, "Detection, estimation and modulation theory," *New York, Wiley*, 1968. (Cité en page 23.)
 - [58] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Transactions on Information Theory*, vol. 45, pp. 1639–1642, 1999. (Cité en page 24.)

-
- [59] J. L. Masson, "Systèmes de transmission avec précodage linéaire et traitement itératif," *Ph.D. dissertation, ELEC-Dépt. Electronique, TELECOM Bretagne*, 2005. (Cité en pages 24 et 25.)
- [60] "Ieee standard for local and metropolitan area networks part 16 : Air interface for fixed and mobile broadband wireless access systems - amendment 2 : Physical layer and medium access control layers for combined fixed and mobile operation in licensed bands," Feb 2006. (Cité en pages 27, 30, 49 et 102.)
- [61] A. Glavieux, C. Laot, and J. Labat, "Turbo equalization over a frequency selective channel," *International Symposium on Turbo codes and Related Topics*, pp. 96–101, 1997. (Cité en page 29.)
- [62] M. Sellathurai and S. Haykin, "Turbo-blast for high speed wireless communications," *IEEE Wireless Communications Networking Conference*, vol. 1, pp. 315–320, 2000. (Cité en page 29.)
- [63] S. M. and H. Simon, "A simplified diagonal blast architecture with iterative parallel-interference cancellation receivers," *IEEE International Conference on Communications*, 2001. (Cité en page 29.)
- [64] M. Sellathurai and S. Haykin, "Turbo-blast for wireless communications : theory and experiments," *IEEE Transactions on Signal Processing*, vol. 50, pp. 2538–2546, 2002. (Cité en page 29.)
- [65] J. Hu and T. Duman, "Graph-based detector for blast architecture," *IEEE International Conference on Communications*, p. 10181023, 2007. (Cité en pages 32, 45 et 46.)
- [66] S. Bavarian and J. Cavers, "A new framework for soft decision equalization in frequency selective mimo channels," *IEEE Transactions on Communications*, vol. 57, pp. 415–422, 2009. (Cité en pages 32 et 46.)
- [67] J. Zhang and M. Fossorier, "Shuffled iterative decoding," *IEEE Transaction on Communications*, vol. 53, pp. 209–213, 2005. (Cité en pages 32, 41, 46 et 102.)
- [68] M. Kaynak, T. Duman, and E. Kurtas, "Belief propagation over MIMO frequency selective fading channels," *International Conference on Networking and Services*, vol. 34, pp. 564–570, 2005. (Cité en pages 35 et 82.)
- [69] X. Yang, Y. Xiong, and F. Wang, "An adaptive MIMO system based on unified belief propagation detection," *IEEE International Conference on Communications*, pp. 209–213, 2007. (Cité en pages 35, 75 et 82.)
- [70] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transaction on Communication*, vol. 49, pp. 1727–1737, 2001. (Cité en pages 42 et 76.)
- [71] —, "Design of serially concatenated codes based on iterative decoding convergence," *Proc. of 2nd International Symposium on Turbo codes*, pp. 3013–322, 2002. (Cité en page 42.)
- [72] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization : principles and new results," *IEEE Transactions on Communications*, vol. 50, pp. 754–767, 2002. (Cité en page 42.)
- [73] C. Hermosilla and L. Szczecinski, "Exit charts for turbo receivers in mimo systems," *International Symposium on Signal processing and ITs applications*, vol. 1, pp. 209–212, 2003. (Cité en pages 42 et 43.)
- [74] J. Yang, C. Nour, and C. Langlais, "Joint factor graph detection for ldpc and stbc coded mimo systems : A new framework," *Process. in Inter. Symp. on Turbo Codes and Iter. Inform.*, pp. 122–126, 2010. (Cité en page 46.)
- [75] A. Haroun, C. Abdel Nour, M. Arzel, and J. Christophe, "Réception itérative mimo basée sur la propagation de croyance et des codes ldpc non-binaires," *GRETSI : 24ème colloque du Groupement de Recherche en Traitement du Signal et des Images*, 2013. (Cité en pages 52 et 115.)
- [76] A. Haroun, C. Abdel Nour, M. Arzel, and C. Jégo, "Symbol-based bp detection for mimo systems associated with non-binary ldpc codes," *IEEE Wireless Communications and Networking Conference*, 2014. (Cité en pages 52 et 115.)

-
- [77] M. Li, C. Nour, C. Jégo, and C. Douillard., “Efficient iterative receiver for bit-interleaved coded modulation according to the dvb-t2 standard,” *IEEE Workshop on Signal Processing Systems*, pp. 18–23, 2009. (Cité en page 59.)
 - [78] A. Haroun, C. Abdel Nour, M. Arzel, and C. Jégo, “Low-complexity ldpc-coded iterative mimo receiver based on belief propagation algorithm for detection,” *8th International Symposium on Turbo Codes and Iterative Information Processing*, 2014. (Cité en pages 62, 79 et 116.)
 - [79] A. Grant, “Convergence of non-binary iterative decoding,” *IEEE Global Telecommunications Conference*, pp. 209–213, 2001. (Cité en page 76.)
 - [80] A. Haroun, C. Abdel Nour, M. Arzel, and C. Jégo, “An efficient mimo receiver based on bp algorithm with truncated message-passing,” *GDR SoC-SiP : 9ème colloque du Groupe de recherche system on chip -system in package*, 2014. (Cité en pages 79 et 116.)
 - [81] —, “Low-complexity layered bp-based detection and decoding for a nb-ldpc coded mimo system,” *IEEE International Conference on Communications*, 2014. (Cité en pages 79 et 116.)
 - [82] A. A. Ghouwayel and E. Boutillon, “Develop the generic vhdl for fpga synthesis,” *DAVINCI Deliverable D6.2.2*, 2010. (Cité en pages 81, 82, 101, 103, 107, 108 et 111.)
 - [83] E. Boutillon, Y. Eustache, P. Bomel, A. Haroune, and L. Conde-Canencia., “Performance measurement of davinci code by emulation,” *DAVINCI Deliverable D6.2.3*, 2010. (Cité en pages 81, 82 et 101.)
 - [84] H. Artes, D. Seethaler, and F. Hlawatsch, “Efficient detection algorithms for mimo channels : a geometrical approach to approximate ml detection,” *IEEE Transaction on Signal Processing*, vol. 51, pp. 2808–2820, 2003. (Cité en page 81.)
 - [85] M. Myllylä, M. Vehkaperä, and M. Juntti, “Complexity evaluation of mmse based detector for lst architectures,” *IEEE International Workshop on Convergent Technologies*, 2005. (Cité en page 81.)
 - [86] K. Wong, C. Tsui, R. Cheng, and W. Mow, “A vlsi architecture of a kbest lattice decoding algorithm for mimo channels,” *IEEE International Symposium on Circuits and Systems*, vol. 3, 2002, pp. 273276. (Cité en page 81.)
 - [87] A. Burg, N. Felber, and W. Fichtner, “A 50 mbps maximum likelihood decoder for multiple-input multiple-output systems with qpsk modulation,” *IEEE International Conference on Electronics, Circuits, and Systems*, vol. 1, 2003. (Cité en page 81.)
 - [88] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Boelskei, “Vlsi implementation of mimo detection using the sphere decoding algorithm,” *IEEE Journal on Solid-State Circuits*, vol. 40, 2005. (Cité en page 81.)
 - [89] C. Studer, A. Burg, and H. Bölskei, “Soft-output sphere decoding : Algorithms and vlsi implementation,” *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 26, 2008. (Cité en page 81.)
 - [90] T. Koike, Y. Seki, H. Murata, S. Yoshida, and A. K., “Fpga prototyping of mimo detector for over-1gbps wireless transmissions,” *XXVIIIth URSI General Assembly*, 2005. (Cité en page 81.)
 - [91] M. Myllylä, J. M. Hintikka, J. R. Cavallaro, M. Juntti, M. Limingoja, and A. Byman, “Complexity analysis of mmse detector architectures for mimo ofdm systems,” *Asilomar, Conference on Signals, Systems and Computers*, pp. 75–81, 2005. (Cité en page 82.)
 - [92] F. Edman and V. Öwall, “A scalable pipelined complex valued matrix inversion architecture,” *IEEE International Symposium on Circuits And Systems*, p. 44894492, 2005. (Cité en page 82.)
 - [93] —, “An fpga implementation of a matrix inversion architecture for multiple antenna algorithms,” *Nordic Radio Symposium*, 2004. (Cité en page 82.)

-
- [94] M. Karkooti, J. R. Cavallaro, and C. Dick, "Fpga implementation of matrix inversion using qdr-rls algorithm," in *Proc. on 2005 Asilomar conference, Pacific Grove*, 2005. (Cité en page 82.)
 - [95] I. LaRoche and A. S. Roy, "An efficient regular matrix inversion circuit architecture for mimo processing," *IEEE International Symposium on Circuits and Systems*, 2006. (Cité en page 82.)
 - [96] W. Chen, C. Poulliat, D. Declercq, L. Conde-Canencia, A. Al-Ghouwayel, and E. Boutillon., "Non-binary ldpc codes defined over the general linear group : Finite length design and practical implementation issues." *IEEE : Vehicular Technology Conference*, pp. 1–5, 2009. (Cité en page 102.)
 - [97] L. Conde-Canencia, A. Al-Ghouwayel, and E. Boutillon, "Propose new decoding algorithm," *DAVINCI Deliverable D6.1.2*,. (Cité en pages 108 et 111.)
 - [98] E. Boutillon and L. Conde-Canencia., "Bubble check : a simplified algorithm for elementary check node processing in extended min-sum non-binary ldpc decoders." *Electronics Letters*, vol. 46, p. 633634, 2010. (Cité en page 108.)
 - [99] —, "Simplified check node processing in nonbinary ldpc decoders." *International Symposium on Turbo Codes and Iterative Information Processing*, pp. 201–205, 2010. (Cité en page 108.)
 - [100] D. Karakolah, "Conception et prototypage dun récepteur itératif pour des systèmes de transmission mimo avec précodage linéaire," *Thèse de doctorat, Télécom Bretagne*, 2009. (Cité en pages 114 et 116.)

Cette thèse présente une solution conjointe de détection et de décodage pour un système multi-antennes basé sur l'algorithme de propagation de croyance. Ce principe de propagation de croyance est appliqué aussi bien à la détection des symboles reçus qu'à leur décodage. Pour ce faire, un code correcteur d'erreurs de type LDPC (Low-Density Parity-Check) non-binaire est exploité. Nous proposons un nouveau schéma de réception itérative qui combine un détecteur MIMO (Multiple-input Multiple-Output) associé à un demapping avec une constellation d'ordre élevé suivi d'un décodeur LDPC non-binaire. Ce détecteur sous-optimal basé sur l'algorithme de propagation de croyance est étudié comme une alternative à un détecteur optimal basé sur le maximum de vraisemblance. L'information extrinsèque est échangée entre le détecteur et le décodeur au sein d'un processus itératif. La méthode des diagrammes EXIT (EXtrinsic Information Transfer) non-binaires a été appliquée afin d'analyser la convergence d'un récepteur itératif. La détection MIMO-BP et le décodage NB-LDPC sont combinés au sein d'un GFC (Graphe à Facteur Conjoint). Ce graphe peut être parcouru selon deux séquençements : le séquençement flooding et le séquençement shuffle. Dans le cadre de cette étude, ces deux types de séquençement sont comparés selon les critères de performance de correction, de latence et de complexité calculatoire.

Le principal objectif de ce travail de thèse est de réduire la complexité calculatoire du récepteur pour permettre son implémentation. Une première étape porte sur la mise à jour des distances euclidiennes du détecteur par les messages a priori provenant du décodeur. Ainsi, nous proposons deux techniques. La première technique porte sur la mise à jour des symboles les plus fiables par rapport au calcul initial des distances euclidiennes du détecteur. La seconde concerne la mise à jour des symboles par rapport aux informations les plus fiables provenant du décodeur. Ces deux propositions permettent une réduction importante de la complexité calculatoire tout en limitant les dégradations au niveau des performances. Dans un second temps, nous proposons une méthode de détection permettant de simplifier le calcul initial des distances euclidiennes. Cette dernière a des performances proches de celles du récepteur initial. Cette méthode consiste à déduire un certain nombre de points à partir du point le plus proche du symbole reçu. Une étude comparative en termes de performance et de complexité calculatoire des détecteurs simplifiés avec le détecteur original montre que les détecteurs proposés ont des caractéristiques favorables à une intégration matérielle au sein d'un système de réception MIMO itératif grâce une complexité divisée par 10 par rapport à la solution de référence.

Enfin, nous proposons une découpe architecturale pour les différents modules constituant le récepteur itératif. L'architecture proposée se compose d'un premier ensemble qui correspond à l'architecture du détecteur MIMO et d'un second pour le décodage du code LDPC-NB. Une représentation architecturale, une organisation temporelle et l'utilisation des ressources arithmétiques et de mémorisation sont détaillées pour les différents modules. Grâce à un niveau de parallélisme approprié, l'architecture associée au récepteur itératif permet un échange d'information efficace et par conséquent la montée en débit et la maîtrise de la latence du récepteur.

Mots-clés : Principe de la propagation de croyance, Codes LDPC non-binaires, Système à multi-antennes, Graphe à facteur conjoint, Information extrinsèque, Récepteur itératif, Diagramme EXIT

In this document, an efficient iterative receiver for digital communication systems is investigated. It combines a Non-Binary Low-Density Parity-Check (NB-LDPC) decoder with a high-order constellation demapper and a Multiple-Input Multiple-Output (MIMO) detector. In the first part of our work, we have proposed a suboptimal MIMO detector based on the Belief Propagation (BP) algorithm as an alternative to a Maximum Likelihood (ML) detector. Extrinsic information is exchanged between the detector and the decoder thanks to an iterative process. Extrinsic Information Transfer (EXIT) charts have enabled to analyze the convergence behaviour of the proposed MIMO-BP detector. A suitable schedule for the different types of iterations is also proposed to reduce the receiver latency and improve its error correction performance. Indeed, a MIMO-BP detection and an NB-LDPC decoding can be combined together to form a larger Joint Factor Graph (JFG) which works on two different schedulings : a flooding scheduling and a Vertical Shuffle Scheduling (VSS). This approach enables to optimize the receiver schedule in terms of inter-iterations and intra-iterations to offer an efficient trade-off between performance and latency.

In a second part, we have focused on the main objective of this work : the reduction of the computational complexity of the iterative receiver to enable its implementation. Actions are undertaken at the detector, decoder and the iterative receiver levels in order to reduce overall computational complexity. The first step involved the updating of Euclidean distances within the detector by a priori messages from the decoder. Thus, we propose to act on two levels : the LDPC decoder level and the Euclidean distance update level. In the first level, a subset of Euclidean distance values is updated by decoder extrinsic information. This subset includes the symbol indexes with highest LLR reliability at the output of the detector during the previous inter-iteration. The second level is the update of the most reliable symbols relative to the initial calculation of the Euclidean distances of the detector. Indeed, applying the proposed low-complexity BP-based detection greatly reduces the number of operations per iteration, with a negligible performance penalty. In a second step, we have proposed a detection method to simplify the initial calculation of Euclidean distances. This method involves reducing a number of points around the closest points of the received symbol. We first calculate the first symbol candidate of the constellation, the remaining Euclidean distances are deduced recursively. This work enables to reduce the number of operations in the detector by a factor of ten when compared to full-complexity BP detection. The main interest of the proposed BP-based detector is to provide a feasible solution for the design of an iterative MIMO receiver.

In the last chapter, we propose a hardware architecture for the iterative receiver which consists in two parts. The first part is the architecture of the MIMO detector and the second part corresponds to the LDPC decoder. An architectural design, a time analysis and the complexity in terms of real operators are detailed for each block of the proposed architecture. Finally, the overall architecture proposed for the iterative receiver enables an efficient information exchange by an increase of the bandwidth and the reduction of the latency.

Keywords : Belief Propagation (BP), Non-Binary Low-Density Parity-Check (NB-LDPC), Multiple-Input Multiple-Output (MIMO), Joint Factor Graph (JFG), Extrinsic Information Transfer (EXIT) chart, Iterative receiver